

(19) 대한민국특허청(KR)  
(12) 등록특허공보(B1)

(51) Int. Cl. <sup>7</sup> G06F 17/30		(45) 공고일자 (11) 등록번호 (24) 등록일자	2002년06월 10일 10-0340169 2002년05월 28일
(21) 출원번호 (22) 출원일자 (73) 특허권자 (72) 발명자 (74) 대리인	10-1999-0041359 1999년09월 27일 줄라이네트 주식회사 서울 강동구 성내1동 468-6 현대코랄빌딩 202호 허홍 서울특별시도봉구쌍문1동523-10 박순은 서울특별시송파구방이동코오롱아파트101동608호 이진배 서울특별시마포구신수동101-8 강희국 서울특별시강동구천호2동463-43 서성철 서울특별시송파구거여동290거여아파트104동804호 이재화	(65) 공개번호 (43) 공개일자	특2001-0028861 2001년04월 06일

심사관 : 임인권

(54) 자바를 이용한 동적 멀티미디어 웹 카타로그 시스템 및 그 방법

요약

본 발명은 서버로부터 웹 브라우저로 스트리밍된 각종 멀티미디어 데이터를 HTML상에 함께 혼합하여 카타로그 내용들을 웹 브라우저로 가장 효과적으로 표현할 수 있는 자바 기반의 동적 멀티미디어 웹 카타로그 시스템 및 그 방법에 관한 것이다.

본 발명은 웹 브라우저를 통해 접속할 때 자바 애플릿 정보가 포함된 HTML 카타로그 문서를 웹 브라우저로 제공하는 웹서버와, 자바 애플릿으로 비디오 데이터를 실시간 상으로 스트리밍해주기 위한 비디오 서버 엔진과, 스트리밍에 적합한 제반정보의 템플레이트 파일을 포함하는 템플레이트 파일시스템과, 멀티미디어 웹 카타로그에 필요한 기능들을 자바 애플릿으로 구현하여 정적인 HTML 문서 내에 삽입하여 멀티미디어 데이터를 서버 시스템에 요청하고, 미디어 데이터들을 동적으로 결합시키는 동기화 엔진과, 카타로그 미디어 데이터 패킷을 스트리밍 방식으로 전달받아 미디어큐에 보관하는 미디어 수신 엔진과, 수신된 미디어 데이터들을 디코딩하고 해당 디바이스로 재생해주는 미디어 플레이어 엔진과, 디코딩된 미디어 데이터를 재생하기 위한 디스플레이 엔진으로 구성되는 것을 특징으로 한다.

대표도

도3

색인어

멀티미디어 웹 카타로그, 자바 애플릿, HTML, 클라이언트 어플리케이션

명세서

도면의 간단한 설명

도 1은 본 발명이 적용되는 일반적인 클라이언트/서버 시스템의 개략 블록도,

도 2는 본 발명에 사용되는 클라이언트 단말기의 블록도,

도 3은 본 발명의 바람직한 실시예에 따른 자바를 이용한 동적 멀티미디어 웹 카타로그 시스템의 블록도,

도 4는 도 3의 자바 애플릿에 포함된 동기화 엔진의 동작을 설명하는 흐름도,

도 5는 도 3의 자바 애플릿에 포함된 미디어 플레이어의 동작을 설명하는 흐름도,

도 6은 도 3의 서버 시스템의 동작을 설명하는 흐름도,

도 7은 도 3의 서버 시스템에 포함된 멀티미디어 템플레이트 파일 시스템의 구성도,

도 8은 도 7의 비디오 템플레이트 파일의 구성도,  
 도 9는 도 3의 자바 애플릿에 포함된 미디어 수신 엔진의 흐름도,  
 도 10은 도 3의 자바 애플릿에 포함된 미디어 디스플레이 엔진의 구성도,  
 도 11은 도 3의 클라이언트 단말기에 포함된 멀티미디어 H/W 장치의 상세 블록도이다.

**\* 도면의 주요부분에 대한 부호설명 \***

C1-Cn ; 클라이언트 단말기 SS1-SSn ; 서버 시스템

12 ; 웹서버 14 ; 웹 브라우저

15 ; 인터넷망 18 ; 자바 버추얼 머신

142 ; 자바 애플릿 1400 ; 비디오 서버 엔진

1402 ; 멀티미디어 템플레이트 파일 시스템

1404 ; 컨텐츠 서버 1420 ; 미디어 수신 엔진

1422 ; 미디어 플레이어 엔진 1424 ; 미디어 큐

1426 ; 동기화 엔진 1428 ; 미디어 디스플레이 엔진

## 발명의 상세한 설명

### 발명의 목적

#### 발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 자바를 이용한 동적 멀티미디어 웹 카타로그 시스템 및 그 방법에 관한 것으로, 특히 정적인 HTML 문서내에 멀티미디어 웹 카타로그에 필요한 기능들을 동적으로 실시간상에 구현하기 위한 자바 애플릿(Java Applet)을 클라이언트 애플리케이션으로 삽입함에 의해, 서버 시스템로부터 웹 브라우저로 스트리밍(Streaming)된 각종 멀티미디어 데이터를 HTML상에 함께 혼합하여 카타로그 내용들을 웹 브라우저로 가장 효과적으로 표현할 수 있는 자바 기반의 동적 멀티미디어 웹 카타로그 시스템 및 그 방법에 관한 것이다.

일반적으로 웹(Web)에서 HTML(Hyper Text Markup Language) 문서 내에 다양한 부가 서비스 및 관련 컨텐츠들(contents)을 포함하여 정적인 정보들을 더욱 다양하고 복합적으로 표현하기 위한 일련의 컨텐츠 표현 방법을 멀티미디어 웹 카타로그(Web Cataloging)(예를들어, 멀티미디어 컨텐츠 표현, 홍보, 배열, 광고 등)이라 하며, 인터넷의 인프라가 기존의 전화선에서 ISDN, ADSL(비동기 디지털 가입자 루프), 케이블 모뎀, 위성 모뎀 등의 다양한 방식으로 확대 되면서 전자상거래(EC)용 웹 쇼핑몰이나 원격 교육 등을 주축으로 그 활용건수가 폭발적으로 증가하고 있다.

이와 같은 멀티미디어 웹 카타로그는 인터넷 백화점이나 쇼핑몰 등과 같이 적은 비용으로 상품을 효과적으로 표현하고 광고, 판매하기를 원하는 전자상거래 유통업체와 원격 멀티미디어 교육을 원하는 가상 대학 등에서 효율적으로 사용할 수 있다.

한편, 동적 멀티미디어 웹 카타로그 시스템은 인터넷상에서 사용자의 요구가 있을 때마다 동적으로 멀티미디어 컨텐츠 및 표현 방법을 구성, 표현 및 관리하는 시스템을 말한다.

통상적으로 멀티미디어 웹 카타로그 시스템을 구축하는 데는 HTML 문서로만 연출하는 경우, 자바 스크립트(JavaScript) 또는 DHTML(Dynamic HTML)을 이용하는 경우, 플러그-인(Plug-In)이나 헬퍼 어플리케이션(Helper Application)을 이용하는 경우로 나누어진다.

먼저, HTML 문서만 사용한 경우는 가장 단순한 형태의 이미지나 내용을 출력하는 형태를 갖는다. 따라서, 서버(Server)에 있는 동적 멀티미디어 내용을 실시간적으로 액세스하여 동적으로 표현하고자 하는 경우나, 복합적인 내용을 동기화하여 출력하고자 하는 경우에는 비효율적이거나 HTML만으로는 구성할 수 없게 된다. 그러므로 별도의 자바 스크립트와 같은 스크립트 프로그램(Script Program)이나 플러그-인 또는 헬퍼 어플리케이션을 설치해야 한다.

자바 스크립트(Java Script)를 사용하는 경우는 프로그램 기능이나 사이즈에 제한이 많고 복잡한 멀티미디어 표현 및 제어가 불가능한 문제가 있다.

또한, DHTML을 이용하는 경우는 간단한 형태의 멀티미디어 제어 및 표현에는 적합하나 보다 지능적인 프로시저(Procedure)를 가진 표현 형태에는 많은 제약점을 가진다. 이 경우 일반 사용자가 구현하기에는 너무 어려우며 또한 시스템의 유지 보수적인 관점에서 볼 때 비효율적이라고 할 수 있다.

별도의 플러그-인(Plug-In)이나 헬퍼 어플리케이션(Helper Application)을 이용하는 경우에는 모든 미디어들을 통합한 형태의 이진 파일로 웹서버에 올려놓고 플러그-인이나 헬퍼 어플리케이션으로 다운로드(download) 실행하기 때문에 기존의 정보를 담고 있는 HTML 문서나 저작 후에 생겨난 정보들과 함께 표현되기 어렵다는 단점을 가지고 있고, 아울러 한 미디어의 정보가 바뀔때 마다 파일 전체를 매번 다시 저작하여 웹서버에 올려놓아야 하는 저작상의 불편함을 가지고 있다.

또한 사용자는 항상 플러그-인이나 헬퍼 프로그램을 자신의 웹 브라우저(Web Browser)에 설치해 놓고 있어야 하며, 이것들은 독자적인 하나의 응용 프로그램이기 때문에 해당 HTML 문서내에 원하는 위치나 형태로 결합될 수 없는 단점을 지니고 있다. 즉, HTML 내에서 레이아웃(layout) 변경 등이 거의 불가능한

문제점이 있다.

대부분의 이러한 프로그램들은 아울러 비디오나 오디오와 같은 연속적 미디어들만을 다루기 위해 제작되어 있기 때문에 이미지, 텍스트, HTML과 같은 비연속적 미디어와 혼합하여 동적으로 재생하는 데에 많은 제작점을 가진다. 또한, 플러그-인이나 헬퍼 프로그램은 모든 클라이언트(Client) 사양마다 다른 버전의 프로그램을 개발해야 하기 때문에 개발 및 유지보수에 대한 보다 복잡한 투자나 비용이 소요되어야 한다.

반면에, 자바(Java)를 이용할 경우 거의 모든 웹 브라우저에서 자동으로 수행될 수 있기 때문에 별도의 버전을 제작할 필요없이 단지 하나의 버전만 웹서버에 올려놓으면 웹 브라우저를 탑재한 모든 클라이언트 시스템에서 구동이 가능하고, 아울러 별도의 어플리케이션(Application) 프로그램을 구성할 수 있기 때문에 스크립트 언어에서 할 수 없었던 보다 지능적인 프로시저어를 구현할 수 있다.

또한, 그 모양이나 사이즈가 HTML 문서내에서 변형이 가능하고 객체지향적 특성에 따라 하나 이상의 객체 복제가 가능해 하나의 HTML 문서내에서 보다 다양하고 혼합적인 멀티미디어 표현이 가능하다. 아울러 헬퍼 프로그램과 같이 독자적인 윈도우로도 실행이 가능하며 자바라는 하나의 통합환경을 통하여 플러그-인이나 헬퍼 프로그램에서 할 수 없었던 다양한 멀티미디어 표현과 하나 이상의 다양한 HTML 문서와도 상호 표현작용이 용이하다.

#### 발명이 이루고자하는 기술적 과제

따라서 본 발명은 이러한 종래기술의 문제점을 감안하여 안출된 것으로, 그 목적은 정적인 HTML 문서내에 멀티미디어 웹 카타로그에 필요한 기능들을 동적으로 실시간상에 구현하기 위한 자바 애플릿을 클라이언트 애플리케이션으로 삽입하고, 서버 시스템로부터 현재 표현에 필요한 각종 멀티미디어 데이터를 각각의 특성에 맞게 동적으로 요청하여 실시간적으로 웹서버를 통해 웹 브라우저로 스트리밍함으로써 HTML상에 함께 혼합하여 카타로그 내용들을 웹 브라우저로 가장 효과적으로 표현하도록 하기 위한 자바 기반의 동적 멀티미디어 웹 카타로그 시스템 및 그 방법을 제공하는데 있다.

본 발명의 다른 목적은 HTML 문서내에 삽입한 자바 애플릿을 클라이언트 프로그램으로 구동하고, 그것의 동기화 엔진과 스트리밍 기능 등을 적절히 혼합하여 실시간상에 선택된 비디오, 오디오, 2D/3D 이미지, 텍스트, 애니메이션, HTML 등의 미디어 프레임들을 웹서버와 비디오 서버 엔진을 통해 논리적으로 송수신/결합시켜 줌으로써, 종래의 HTML이나 DHTML(Dynamic-HTML) 문서, 플러그-인(Plug-In) 프로그램(S/W), 자바 스크립트로 표현하기 어려웠던 동적 멀티미디어 웹 카타로그 정보들을 웹 브라우저 내에 효과적으로 표현해 줄 수 있는 자바 기반의 동적 멀티미디어 웹 카타로그 시스템 및 그 방법을 제공하는데 있다.

본 발명의 또 다른 목적은 하드웨어에 독립적인 순수 자바(Pure-Java) 기반으로 설계되어 자바 버추얼 머신(Java Virtual Machine)을 탑재한 PC(개인용 컴퓨터)나 워크스테이션, 노트북 PC, 팜탑 PC의 웹 브라우저에서는 물론, 퍼스널 자바(Personal Java)를 포팅(porting)한 일반 휴대용 통신단말기(PCS, PHS), 휴대 정보 단말기(PDA), 셋톱 박스(settop box), 디지털 텔레비전(Digital TV), 웹폰(Web Phone) 및 자바 오퍼레이팅 시스템(JavaOs)을 탑재한 자바 칩(Java Chip) 등에서 실행이 가능한 자바 기반의 동적 멀티미디어 웹 카타로그 시스템 및 그 방법을 제공하는데 있다.

본 발명의 다른 목적은 스트리밍 기반으로 이루어져 있기 때문에 인터넷의 대역폭에 따라 필요한 미디어들만 선택적으로 받아 볼 수 있고, 아울러 프레임 기반의 가상 VCR을 이용한 지능적 재생이 가능한 자바 기반의 동적 멀티미디어 웹 카타로그 시스템 및 그 방법을 제공하는데 있다.

본 발명의 또 다른 목적은 종래의 비디오, 오디오 등의 연속적 미디어 스트리밍 방법에서 이미지, 텍스트, HTML 등의 비연속적 미디어 스트리밍 방식을 새롭게 고안하여 웹상에서의 가장 적합한 멀티미디어 표현 방식을 제공할 수 있는 자바 기반의 동적 멀티미디어 웹 카타로그 시스템 및 그 방법을 제공하는데 있다.

#### 발명의 구성 및 작용

상기한 목적을 달성하기 위하여, 본 발명은 웹 브라우저와 자바 버추얼 머신을 포함하는 클라이언트 단말기와 웹서버를 포함하는 서버 시스템이 인터넷망을 통하여 연결된 클라이언트/서버 시스템 환경에서 자바를 이용한 동적 멀티미디어 웹 카타로그 시스템에 있어서, 비디오, 오디오, 2D/3D 이미지, 애니메이션, 텍스트 및 HTML을 포함하는 멀티미디어 데이터를 스트리밍에 적합하도록 제반정보를 포함한 템플레이트 파일로 이루어진 멀티미디어 템플레이트 파일시스템과; 사용자가 웹 브라우저를 통해 접속할 때 클라이언트 자바 애플릿 정보가 포함된 HTML 카타로그 문서를 웹 브라우저로 제공하며, 상기 클라이언트 자바 애플릿 정보에 따라 웹 브라우저에 생성된 클라이언트 자바 애플릿의 요구가 있는 경우 오디오, 2D/3D 이미지, 애니메이션, 텍스트 및 HTML을 포함하는 멀티미디어 데이터에 대하여는 직접 멀티미디어 템플레이트 파일시스템을 액세스하여 클라이언트 자바 애플릿으로 스트리밍 해주는 웹서버와; 상기 클라이언트 단말기로부터의 비디오 데이터에 대한 요구에 따라 그 내용을 해석하여 웹서버를 통하여 클라이언트 자바 애플릿으로 비디오 데이터를 실시간 상으로 스트리밍해주기 위한 비디오 서버 엔진과; 상기 웹서버에 접속할 때 HTML 카타로그 문서 내에 삽입되어 웹 브라우저로 다운로드되어 자바 버추얼 머신에 의해 생성되며 멀티미디어 웹 카타로그에 필요한 기능들을 동적으로 실시간 상에 구현할 수 있는 클라이언트 자바 애플릿으로 구성되며, 상기 클라이언트 자바 애플릿은 클라이언트의 웹 브라우저 상에서 해당 논리적 시각에 필요한 비디오를 포함한 멀티미디어 데이터를 서버 시스템에 요청하고, 웹서버를 통해 전달받은 멀티미디어 데이터를 동적으로 결합하여 재생시키기 위한 동기화 엔진과, 상기 웹서버와 비디오 서버 엔진으로부터 카타로그 멀티미디어 데이터 패킷을 HTTP를 통하여 스트리밍 방식으로 전달받아 미디어큐에 임시로 저장하는 미디어 수신 엔진과, 상기 동기화 엔진의 시간정보, 재생정보, 속도정보 등의 가상 VCR 정보를 계속 체크하면서 해당되는 논리적 시각에 해당 미디어 데이터를 미디어큐로부터 읽어들이어 각 파일 포맷에 맞게 디코딩하고 해당 장치 디바이스로 재생되도록 제어하는 미디어 플레이어 엔진과, 상기 미디어 플레이어 엔진에서 디코딩된 미디어 데이터를 재생하기 위한 디스플레이 엔진으로 구

성되는 것을 특징으로 하는 자바를 이용한 동적 멀티미디어 웹 카타로그 시스템을 제공한다.

본 발명의 다른 특징에 따르면, 본 발명은 브라우저와 자바 버추얼 머신을 포함하는 클라이언트 단말기와, 웹서버를 포함하는 서버 시스템이 인터넷망을 통하여 연결된 클라이언트/서버 시스템 환경에서 자바를 이용한 동적 멀티미디어 웹 카타로그 방법에 있어서, 상기 웹 브라우저를 사용하여 서버 시스템의 웹 서버에 접속하는 단계와, 상기 웹 서버로부터 클라이언트 자바 애플릿 정보가 포함된 HTML 카타로그 문서를 자신의 웹 브라우저로 다운 받는 단계와, 상기 웹 브라우저에서 HTML 카타로그 문서의 HTML 태그를 파싱한 후 자바 버추얼 머신을 통해 멀티미디어 웹 카타로그에 필요한 비디오, 오디오, 이미지, 텍스트, HTML 데이터 중 적어도 하나 이상을 포함하는 멀티미디어 데이터를 서버 시스템에 요청하고, 웹서버를 통해 전달받은 멀티미디어 데이터를 동적으로 결합하여 재생시키기 위한 동기화 엔진과 서버 시스템으로부터 웹 카타로그 미디어의 데이터 패킷을 스트리밍 방식으로 전달받아 미디어큐에 보관하는 미디어 수신 엔진을 포함하는 클라이언트 자바 애플릿 객체를 생성하고 정해진 HTML 문서 내에 GUI를 생성하는 단계와, 그 후 사용자의 초기 입력값이 웹 브라우저를 통하여 입력되면 사용자 입력 파라미터를 파싱하여 사용자가 원하는 카타로그 표현을 위해 선택한 미디어들에 대한 플래그를 설정하는 단계와, 상기 각 미디어에 대한 표현정보가 설정되면 설정된 정보에 기초하여 선택된 미디어들에 대응하여 각각 수신받은 각각의 미디어 데이터를 각 파일 포맷에 맞게 디코딩하고 해당 장치 디바이스로 재생해주는 비디오 플레이어, 오디오 플레이어, 이미지 플레이어, 텍스트 플레이어 및 HTML 플레이어를 포함하는 미디어 플레이어 객체를 생성함과 동시에 각각의 미디어 플레이어 객체에 대응하는 미디어큐를 하나씩 퍼블릭 타입으로 할당받는 단계와, 상기 미디어 플레이어가 생성된 경우 클라이언트 자바 애플릿은 카타로그 표현을 시작하기 위한 초기화 입력 데이터를 구성하여 자바 이벤트 핸들러에 넘겨주는 단계와, 상기 파싱된 입력값에 기초하여 속도조절, 재생위치 및 재생동작 선택에 따른 설정된 속도값과, 설정된 재생위치와, 설정된 초당 프레임율(FPS)을 동기화 엔진으로부터 웹서버를 통해 서버 시스템으로 전달하는 단계와, 상기 요구에 따라 비디오 데이터에 대한 요구는 비디오 서버 엔진이 그 내용을 해석하여 웹서버를 통해 클라이언트 자바 애플릿으로 스트리밍해주며, 비디오 데이터 이외의 오디오, 이미지, 텍스트, HTML 데이터들은 웹서버가 직접 멀티미디어 템플레이트 파일 시스템으로부터 읽어 들어 클라이언트 자바 애플릿으로 스트리밍해주는 단계와, 상기 미디어 수신 엔진에 의해 카타로그 미디어 데이터 패킷을 스트리밍 방식으로 수신하여 미디어큐에 보관하는 단계와, 수신받은 각각의 미디어 데이터를 동기화 엔진의 재생시간 제어에 따라 동기화가 이루어지면서 미디어 플레이어에 의해 각 파일 포맷에 맞게 디코딩하여 해당하는 장치 디바이스로 재생해주는 단계와, 상기 미디어 플레이어에서 디코딩된 미디어 데이터를 디스플레이 엔진에 의해 재생하는 단계로 구성되는 것을 특징으로 하는 자바를 이용한 동적 멀티미디어 웹 카타로그 방법을 제공한다.

상기한 바와같이 본 발명에서는 정적인 HTML 문서내에 멀티미디어 웹 카타로그에 필요한 기능들을 동적으로 실시간상에 구현하기 위한 자바 애플릿을 클라이언트 애플리케이션으로 삽입하고, 서버 시스템에 현재 표현에 필요한 각종 멀티미디어 데이터를 각각의 특성에 맞게 동적으로 요청하여 실시간으로 웹서버를 통해 웹 브라우저로 스트리밍함으로써 HTML상에 함께 혼합하여 카타로그 내용들을 웹 브라우저로 가장 효과적으로 표현할 수 있다.

또한, 본 발명에서는 HTML 문서내에 삽입한 자바 애플릿을 클라이언트 프로그램으로 구동하고, 그것의 동기화 엔진과 스트리밍 기능 등을 적절히 혼합하여 실시간상에 선택된 비디오, 오디오, 2D/3D 이미지, 텍스트, 애니메이션, HTML 등의 미디어 프레임들을 웹서버와 상기 비디오 서버 엔진을 통해 논리적으로 송수신/결합시켜 줌으로써, 종래의 HTML이나 DHTML(Dynamic-HTML) 문서, 플러그-인(Plug-In) 프로그램(S/W), 자바 스크립트로도 표현하기 어려웠던 동적 멀티미디어 웹 카타로그 정보들을 웹 브라우저 내에 효과적으로 표현해 줄 수 있게 되었다.

더욱이, 본 발명 시스템은 하드웨어에 독립적인 순수 자바(Pure-Java) 기반으로 설계되어 자바 버추얼 머신(Java Virtual Machine)을 탑재한 PC(개인용 컴퓨터)의 웹 브라우저에서는 물론, 퍼스널 자바(Personal Java)를 포팅(porting)한 일반 휴대용 통신단말기(PCS, PHS), 휴대 정보 단말기(PDA), 셋톱 박스(settop box), 디지털 텔레비전(Digital TV), 웹폰(Web Phone) 및 자바 오퍼레이팅 시스템(JavaOS)을 탑재한 자바 칩(Java Chip) 등에서 실행이 가능하다.

또한, 본 발명은 스트리밍 기반으로 이루어져 있기 때문에 인터넷의 대역폭에 따라 필요한 미디어들만 선택적으로 받아 볼 수 있고, 아울러 프레임 기반의 가상 VCR을 이용한 지능적 재생이 가능하며, 종래의 비디오, 오디오 등의 연속적 미디어 스트리밍 방법에서 이미지, 텍스트, HTML 등의 비연속적 미디어 스트리밍 방식을 새롭게 고안하여 웹상에서의 가장 적합한 멀티미디어 표현 방식을 제공할 수 있다.

#### (실시예)

이하에 상기한 본 발명을 바람직한 실시예가 도시된 첨부도면을 참고하여 더욱 상세하게 설명한다.

첨부된 도 1은 본 발명이 적용되는 일반적인 클라이언트/서버 시스템의 개략 블록도, 도 2는 본 발명에 사용되는 클라이언트 단말기의 블록도, 도 3은 본 발명의 바람직한 실시예에 따른 자바를 이용한 동적 멀티미디어 웹 카타로그 시스템의 블록도, 도 4는 도 3의 자바 애플릿에 포함된 동기화 엔진의 동작을 설명하는 흐름도, 도 5는 도 3의 자바 애플릿에 포함된 미디어 플레이어의 동작을 설명하는 흐름도, 도 6은 도 3의 서버 시스템의 동작을 설명하는 흐름도, 도 7은 도 3의 서버 시스템에 포함된 멀티미디어 템플레이트 파일 시스템의 구성도, 도 8은 도 7의 비디오 템플레이트 파일의 구성도, 도 9는 도 3의 자바 애플릿에 포함된 미디어 수신 엔진의 흐름도, 도 10은 도 3의 자바 애플릿에 포함된 미디어 디스플레이 엔진의 구성도이다.

먼저, 도 1은 본 발명이 적용되는 일반적인 클라이언트/서버 시스템으로서 각각 웹 브라우저(14)와 자바 버추얼 머신(JVM: Java Virtual Machine)(18)을 탑재한 다수의 클라이언트 단말기(C1-Cn)와 각각 웹서버(12)를 포함한 다수의 서버 시스템(SS1-SSn)이 인터넷(15)을 통하여 상호 연결되어 있다.

상기한 각 클라이언트 단말기는 예를들어, 도 2에 도시된 바와같이 멀티미디어 처리기능과 유선 또는 무선으로 인터넷 접속이 가능하도록 통신기능을 구비한 데이터 처리장치이면 모두 사용이 가능하다.

즉, 클라이언트 단말기(C1)는 중앙처리장치(CPU)와(2), 메인 메모리(3)를 포함하고 있다. CPU(2)와 메인 메모리(3)는 버스(BUS)(10)를 통하여 하드 디스크 등으로 구성되는 보조기억장치(4) 및 ROM(5)에 접속되어 있다.

또한, 보조기억장치(4)와 ROM(5)에는 오퍼레이팅 시스템(OS)과 협동으로 CPU(2) 등에 명령을 부여하기 위한 프로그램 코드가 기록되어 있으며, 메인 메모리(3)에 로드됨에 따라 실행된다.

클라이언트 단말기(C1)는 사용자 인터페이스를 위해 키보드/마우스와 같은 입력장치(7)와, CRT 또는 LCD 모니터나 빔 프로젝터와 같은 표시장치(8)와, 스피커와 같은 음성출력장치(9)가 각각의 컨트롤러(도시되지 않음)를 통하여 내부 버스(BUS)(10)에 접속되어 있다.

더욱이, 클라이언트 단말기(C1)는 서버 시스템(SS1)과의 통신을 위하여 통신용 모뎀, PCMCIA 네트워크 카드 또는 이더넷 네트워크 카드와 같은 통신 인터페이스(6)를 구비하며, 보조기억장치(4)에는 이들을 통하여 인터넷망(15)에 접속된 서버 시스템(SS1)에 접속하는데 필요한 프로그램으로서 웹 브라우저(14)와 자바 버추얼 머신(JVM)(18)이 기억되어 있고, 인터넷 접속시에 메인 메모리(3)로 로드되어 구동된다.

상기한 바와같이 클라이언트 단말기(C1)에 요구되는 기능은 유선 또는 무선으로 인터넷 접속이 가능한 통신기능과 서버 시스템(SS1)으로부터 수신한 멀티 미디어 데이터를 재생하는데 필요한 멀티미디어 처리 기능을 갖는 단말기이면 어떤 것도 적용이 가능하다.

따라서, 클라이언트 단말기(C1)로는 자바 버추얼 머신(JVM)을 탑재한 웹 브라우저, 즉 자바로 구동되는 웹 브라우저(14)가 인스톨되어 있는 데스크탑 컴퓨터에서는 물론, 퍼스널 자바(PersonalJava)를 포팅(porting)한 일반 휴대용 통신단말기(PCS, PHS 등), 휴대 정보 단말기(PDA), 셋톱 박스(settop box) 디지털 텔레비전(Digital TV), 웹폰(Web Phone) 및 자바 오퍼레이팅 시스템(JavaOs)을 탑재한 자바 칩(Java Chip) 등이 사용될 수 있다.

한편, 본 발명의 서버 시스템(SS1)은 도 3에 도시된 바와같이 크게 인터넷망(15)에 접속된 웹서버(12)와, 웹서버(12)에 접속되어 클라이언트 단말기(C1)로부터의 비디오 데이터에 대한 요구(request)에 따라 그 내용을 해석하여 웹서버(12)를 통하여 클라이언트 자바 애플릿(142)으로 비디오 데이터를 실시간 상으로 스트리밍해주기 위한 비디오 서버 엔진(1400)과, 웹서버(12)에 접속되며 클라이언트 단말기(C1)로부터의 요구(request)에 따라 웹서버(12)를 통하여 클라이언트 자바 애플릿(142)으로 멀티미디어 데이터 스트리밍에 적합하도록 설계된 템플레이트 파일들로 이루어진 멀티미디어 템플레이트 파일시스템(1402)을 포함한다.

즉, 상기 템플레이트 파일은 비디오 이외의 오디오, 2D/3D 이미지, 애니메이션, 텍스트 멀티미디어 데이터를 스트리밍에 적합하도록 제반정보(웹서버내 저장위치, 매체타입, 매체이름, 표현시작시간, 표현 끝 시간 등)들을 담고 있다.

더욱이, 상기 멀티미디어 템플레이트 파일시스템(1402)의 미디어 파일 디렉토리는 후술하는 바와같이 TCP/IP(Transmission Control Protocol/Internet Protocol)를 기반으로 적어도 일 이상의 콘텐츠 서버(1404)와 상호 물리적인 저장위치를 공유할 수 있다.

상기한 서버 시스템(SS1)은 사용자가 웹 브라우저(10)를 사용하여 웹서버(12)에 접속할 때 웹서버(12)를 통하여 후술하는 도 3에 도시된 바와같은 자바 애플릿(142) 정보가 포함된 HTML 카탈로그 문서를 제공한다.

도 7은 서버 시스템(SS1)의 멀티미디어 템플레이트 파일 시스템(1402)에 대한 구성도로서, 멀티미디어 템플레이트 파일 시스템(1402)은 크게 물리적으로 템플레이트 파일 디렉토리(140202)와 미디어 파일 디렉토리(140204)로 나뉘어진다.

템플레이트 파일 디렉토리(140202)는 실제 미디어 데이터를 핸들링할 메타 데이터(metadata)로 구성된 디렉토리로서, 비디오 템플레이트 파일(140206), 이미지 템플레이트 파일(140208), 텍스트 템플레이트 파일(140210), HTML 템플레이트 파일(140212)로 구성되어 있다.

상기 비디오 템플레이트 파일은 도 8의 구성도와 같이, 크게 나누어서 종래의 MPEG-1 비디오 파일의 입/출력(10) 속도를 스트리밍에 적합하도록 바꾸기 위해 GlobalInfoHeader(1402061), MPEGHeader(1402062), MPEG data stream(1402063), OFFSET\_PER\_FRAME(1402064), TIME\_PER\_FRAME(1402065)의 조각들로 구성되어진다.

이러한 비디오 템플레이트 파일(140206)은 서버가 초기화 될 때 서버 시스템(SS1)의 메모리로 자동 로드되기 때문에 실제 클라이언트 단말기(C1)로부터의 요구(request)가 도착했을 때 빠르게 스트리밍을 시작할 수 있다.

비디오 템플레이트 파일(140206) 이외의 다른 미디어 템플레이트 파일들은 다음과 같은 구조로 되어 있다: {미디어를 가져올 미디어 파일 디렉토리, 미디어 타입, 논리적 시작 시간, 논리적 끝 시간}.

여기서 미디어 타입은 파일의 확장자로 인식한다. 예를 들어, gif/jpeg 등은 이미지 타입, txt는 텍스트, html/htm은 HTML 문서 타입을 나타낸다. 본 템플레이트 파일은 사용자가 텍스트 기반의 에디터로 작성할 수 있다. 본 발명에서는 종래의 방식과 다르게 모든 미디어들의 내용을 하나의 이진 파일에 통합해 놓지 않았기 때문에 언제나든 변경 사항의 수정이 용이하다.

상기한 미디어 파일 디렉토리(140204)는 실제 미디어 데이터들을 저장하고 있는 물리적인 파일 시스템으로서 비디오 파일(140214), 오디오 파일(140216), 이미지 파일(140218), 텍스트 파일(140220) 및 HTML 파일(140222)을 포함한다.

또한, 미디어 파일 디렉토리(140204)는 상기한 바와같이 일 이상의 콘텐츠 서버(1404)와 TCP/IP를 기반으로 하여 상호 물리적인 저장위치를 공유할 수 있으며, 서버 시스템의 성능 및 네트워크의 부하를 고려하여 특정 미디어 데이터를 특정 콘텐츠 서버(1404)에 분산 배치시킬 수 있도록 설계된다.

따라서, 사용자로부터는 하나의 논리적 공간으로 특정 매체를 검색할 수 있도록 하나 실제 그 이면에는 하나 이상의 다른 네트워크로 연결된 물리적인 저장위치에 놓일 수 있게 된다. 이는 상기한 바와같이 서버 시스템의 성능 및 네트워크의 부하를 낮게 유지함과 동시에 제한된 저장공간을 다수의 콘텐츠 서버로 분산시킴에 따라 공간제약의 단점을 극복할 수 있게 된다.

도 3에 도시된 본 발명에 따른 동적 멀티미디어 웹 카탈로그 시스템의 구동은 처음 사용자가 웹 브라우저(14)를 통해 웹서버(12)에 접속하여 클라이언트 자바 애플릿(Java Applet)(142) 정보가 포함된 HTML 카탈로그 문서를 자신의 웹 브라우저(14)로 다운 받으면서 시작된다. 그 문서의 HTML 태그가 웹 브라우저(14)에서 파싱(parsing)되면서 애플릿 태그를 만나는 순간 웹 브라우저(14)는 하나의 클라이언트 자바 애플릿(142) 객체를 자신에 내장된 자바 버추얼 머신(JVM)(18)을 통해 생성하고, 정해진 HTML 문서내의 위치에 GUI(Graphical User Interface)를 생성한다.

이렇게 생성된 클라이언트 자바 애플릿(142)은 정적인 HTML(HyperText Markup Language) 문서 내에 멀티미디어 웹 카탈로그(멀티미디어 콘텐츠 표현, 홍보, 배열, 광고 등)에 필요한 기능들을 동적으로 실시간 상에 구현할 수 있는 자바 애플릿으로 구현하고 삽입하여 서버 시스템(SS1)과 함께 구동하되, 실행시 자바 버추얼 머신(JVM)(18)을 탑재한 클라이언트의 웹 브라우저(14) 상에서 해당 논리적 시각에 필요한 비디오, 오디오, 텍스트, 2D/3D 이미지, 애니메이션, HTML 등의 멀티미디어 데이터를 서버 시스템(SS1)에 게 요청하고, 웹서버(12)를 통해 전달받은 미디어 데이터들을 동적으로 결합하여 재생시킬 수 있는 동기화 로직을 포함하는 동기화 엔진(1426)과; 상기 웹서버(12)와 비디오 서버 엔진(1400)으로부터 카탈로그 미디어 데이터(비디오, 오디오, 이미지, 텍스트, HTML) 패킷을 HTTP(HyperText Transfer Protocol)를 통하여 스트리밍 방식으로 전달받아 미디어큐(Queue)(1424)에 보관하는 미디어 수신 엔진(1420); 수신받은 각각의 미디어 데이터들을 각 파일 포맷에 맞게 디코딩하고 해당 장치 디바이스로 재생해주는 미디어 플레이어 엔진(1422)과; 상기 미디어 플레이어 엔진(1422)에서 디코딩된 미디어 데이터를 재생하기 위한 디스플레이 엔진(1428)으로 구성된다.

상기 동기화 엔진(1426)의 역할을 좀더 상세하게 설명하면 상기 HTML내에 삽입된 자바 애플릿(142)으로부터 멀티미디어 혼합에 관한 정보를 전달받아 선택된 미디어들에 대하여 웹서버(12)로부터 오는 각각을 논리적 시간상에 동기화하여 각 표현 시간을 관리해 주는 수명(Lifecycle)관리 역할과 웹 카탈로그에 관련된 제반정보(웹서버(142) 내의 저장위치, 매체타입, 매체이름, 표현시작시간, 표현 끝시간 등)를 담고 있는 템플레이트 파일을 실시간적으로 해석하여 그 결과로서 상기 자바 애플릿(142)과 일이상의 HTML 프레임(Frame)내에 그 객체들을 동적으로 표현하고 관리하는 서비스 매니저 역할을 한다.

여기에서 웹서버(12)로부터 전달되는 각 미디어 데이터들은 모두 프레임 단위로 존재하며, 각각의 프레임은 실제 표현되어질 이진 압축 데이터, 논리적 표현 시작 시간 및 논리적 표현 끝 시간 등의 속성들로 구성된다.

상기 미디어 플레이어 엔진(1422)은 비디오 플레이어(14220), 오디오 플레이어(14222), 이미지 플레이어(14224), 텍스트 플레이어(14226) 및 HTML 로더(14228)로 구성되어 있다.

상기와 같이 생성된 클라이언트 자바 애플릿(142)은 그후 웹 브라우저(10)를 통해 사용자 입력을 받아 자신의 동기화 엔진(1426)으로부터 해당 요구(request)를 발생시켜 웹서버(12)를 통해 서버 시스템(140)으로 그 요구를 전달하는데, 비디오 데이터에 대한 요구는 비디오 서버 엔진(1400)이 그 내용을 해석하여 웹서버(12)를 통해 클라이언트 자바 애플릿(142)으로 스트리밍해주며, 그외의 오디오, 이미지, 텍스트, HTML 데이터들은 웹서버(12)가 직접 멀티미디어 템플레이트 파일 시스템(1402)로부터 읽어 들여 클라이언트 자바 애플릿(142)으로 스트리밍 해준다.

이렇게 미디어 스트리밍을 두 가지로 나눈 이유는 시스템 부하가 많이 걸리는 비디오에는 전문적인 프로세스를 할당하여 서버 시스템쪽 스트리밍 알고리즘을 특화시켜 적용하고, 그 이외의 상대적으로 가벼운 미디어들은 웹서버(12)의 기본 기능을 이용해 스트리밍 해줌으로써 상대적인 시스템 부하를 최소화하며 서버 시스템을 운영하고자 함에 있다.

이렇게 해서 클라이언트 자바 애플릿(142)으로 전달된 각 멀티미디어 데이터들은 미디어 수신 엔진(1420)을 통해 미디어 큐(1424)에 임시로 저장되며, 각 미디어 플레이어(14220-14228)들은 동기화 엔진(1426)의 시간 정보 및 재생 정보, 속도 정보 등의 가상 VCR 정보를 계속 체크하면서 해당되는 논리적 시각에 해당 미디어 데이터를 미디어 큐(1424)로부터 읽어 들여(get), 자체 디코더를 통해 디코딩 한 후 미디어 디스플레이 엔진(1428)의 해당 미디어 장치를 통해 웹 브라우저(14)로 출력한다.

한번 출력된 각 미디어 데이터들은 자신의 논리적 재생시간 만큼만 존속하며, 그 시간이 만료된 이후에는 해당 미디어 플레이어(14220-14228)에 의해 미디어 큐(1424)에서 제거(remove)된다. 이렇게 함으로써 메모리 관리를 효율적으로 수행 할 수가 있다.

한편, 각 해당 논리적 시간 내에 재생되는 미디어들은 비디오, 오디오, 텍스트, 2D 스틸 이미지, 3D 모델링을 위한 2D 이미지 세트(set), 애니메이션 이미지, HTML 문서들로 구분되며, 이들은 클라이언트 자바 애플릿(140)내 및 다른 웹 브라우저(14) 창에서 서로의 논리적 재생 시간에 독립적으로 수행되어진다.

3D 모델링을 위한 2D 이미지 세트를 제외한 다른 모든 미디어들은 독립적인 재생시간 단위로 쪼개어진 프레임들로 구성된다. 반면에 3D 모델링을 위한 2D 이미지 세트는 이미지 플레이어(14224)에서 한편의 3D 모델링 이미지 및 애니메이션을 만들기 위한 독립적인 형태로 이용된다.

상기와 같은 미디어들의 전반적인 실용적인 예를 들면, 만약 비디오와 자막이 동시에 재생되는 경우, 비디오의 각 데이터 프레임들은 frames/sec 즉, 초당 프레임수를 기준으로 자신의 논리적 재생시간이 결정되며, 자막 텍스트, 스틸 이미지와 같은 경우에는 한 문장이나 그 내용이 비디오 내용과 일치하는 순간 동안 재생시간을 유지한다. 반면에, 오디오 데이터 같은 경우는 자신의 초당 샘플링율(sample/sec)에 따라 각 데이터 프레임들의 재생시간이 결정된다.

이러한 개념들에 따라 멀티미디어 템플레이트 파일 시스템(1402) 내에서는 각 미디어 데이터에 대한 제

반정보(웹서버내 저장위치, 매체타입, 매체이름, 표현시작시간, 표현 끝시간 등)를 사용자가 일반 텍스트 에디터로 쉽게 입력할 수 있도록 텍스트 기반의 파일로 저장하며, 이 템플레이트 파일은 클라이언트 자바 애플릿(140)이 처음 초기화 할 때, 동기화 엔진(1426)이 참조하여 기본 동기화 전략을 수립하는 데에 사용된다.

아울러 비디오 데이터를 제외한 다른 미디어 데이터들은 멀티미디어 템플레이트 파일 시스템(1402) 내에 제반정보와 물리적으로 분리된 채로 함께 저장되며, 각 미디어 플레이어(14220-14228)들에 의해서 독립적으로 스트리밍이 수행된다.

만약 3D 모델링을 위한 2D 이미지 세트를 함께 사용한다면 상기에서 언급했듯이 데이터 단위 형태가 틀리기 때문에 이 부분만 독자적으로 이미지 플레이어(14224)에서 수행된다.

상기와 같은 구조로 웹 카타로그를 동적으로 수행할 때 다음과 같은 미디어들의 조합이 가능해지며, 이러한 미디어 조합들은 각각의 독자적인 동기화 정보들을 갖고 운영되기 때문에 보다 다양하고 복합적인 멀티미디어 재생 모델을 제시할 수 있다. 각 미디어들은 독자적으로도 수행될 수 있으며 하나 이상의 같은 미디어 객체나 다른 미디어 객체들과 연동되어서 재생될 수 있다.

- 비디오 + 텍스트 자막
- 비디오 + 오디오
- 비디오 + 오디오 + 텍스트 자막
- 비디오 + 오디오 + 텍스트 자막 + 스틸 이미지
- 비디오 + 오디오 + 텍스트 자막(한글) + 텍스트 자막(영문)
- 비디오 + 오디오 + 텍스트 자막(한글) + 텍스트 자막(영문) + 텍스트 자막(일본어) + ...
- 비디오 + 오디오 + HTML 문서
- 비디오 + 오디오 + 텍스트 자막 + HTML 문서
- 비디오 + 오디오 + 텍스트 자막 + HTML 문서 + 스틸 이미지
- 스틸 이미지 + 텍스트 자막 => 슬라이드 쇼
- 스틸이미지 + 텍스트 자막 + 각 이미지 당 오디오 => 오디오 슬라이드 쇼
- 애니메이션 이미지 => 롤링 배너광고
- 애니메이션 이미지 + 각 이미지 당 오디오 => 사운드 롤링 배너 광고
- 3D 파노라마 이미지
- 다른 각도에서 찍은 2D 이미지의 연속 => 3D 사진 모델링 효과

특히 HTML 동기화에 있어서 다양한 HTML 프레임을 활용할 경우에 특정 윈도우 이름을 만들어서 호출할 수 있도록 자바 애플릿(142) 내에 HTML 로더(14228)가 구성되어 있기 때문에 보다 다양한 형태의 복합문서 형태를 유지할 수 있어 웹 카타로그에 매우 효율적이다.

이하에 서버 시스템(SS1)과 클라이언트 자바 애플릿(142)을 서로 분리하여 그 동작 원리를 보다 자세히 살펴보면 다음과 같다.

우선 클라이언트 자바 애플릿(142)내의 동작을 살펴보면, 사용자의 초기 입력에서부터 구동이 되기 시작하는데 사용자 초기입력은 HTML 내에 삽입된 태그나 파라미터 형태로 전달받을 수 있다. 또는 자바 스크립트(JavaScript)를 통해 HTML내의 특정 오브젝트에서 자바 애플릿(142)내의 퍼블릭 메소드(Public Method)들을 호출하여 입력값을 전달할 수 있다.

우선, 가장 일반적인 경우로 가정하여 HTML내의 파라미터 값을 통하여 초기입력 값이 전달된다고 했을 때, 이러한 입력들은 주로 클라이언트 자바 애플릿(142)의 GUI 생성정보, 특정 미디어 선택, 재생속도 설정, 초기 임의의 재생위치(Random Access), 가상 VCR 기능(PLAY, STOP, PAUSE, FF, REW), 동기화될 미디어들에 대한 제반정보(미디어 타입, 서버 위치, 파일명, 총 프레임 수, 각 프레임 당의 재생시간 혹은 이러한 정보들을 담고 있는 템플레이트 파일의 위치 등)가 될 것이다. 이러한 초기 입력값들은 동기화 엔진(1426)을 거쳐 필터링되고 이것에 의해 대응액션이 취해진다.

도 4에는 동기화 엔진(1426)의 동작을 설명하는 흐름도가 도시되어 있다.

도 4를 참조하면, 우선 클라이언트 자바 애플릿(142)이 HTML내에서 초기화 될 때, 퍼블릭(Public) 타입의 동기화 엔진(1426) 객체를 하나의 독립적인 스레드(thread)로 생성시키고 대몬 프로세스와 같이 자동수행을 시키며 사용자 입력값이 생길 때마다 이에 반응하도록 만든다(S1426).

이어서, 사용자 초기 입력값이 전달이 되면(S142600), 입력값이 웹 브라우저(14)로부터 입력되었는지를 판단하고(S142602), 만약 입력값이 웹 브라우저(14)로부터 입력된 경우 사용자 입력 파라미터를 파싱하고(S142606), 우선적으로 사용자가 본 카타로그 표현을 위해 선택한 미디어들에 대한 플래그들을 세팅한다(S142608).

만약 비디오가 선택되었다면 클라이언트 자바 애플릿(142)의 전역변수인 'video\_selected' 플래그를 'ON'값으로 설정한다. 다른 미디어들도 마찬가지로 과정을 거친다. 만약 이미지 미디어가 선택되었다고 하면 이것의 타입이 'JPEG'인지, 'GIF'인지, 'Animated GIF'인지에 대한 정보를 추가로 파악한다. 만약 아무런 사항이 입력되지 않았다면 기본적으로 모든 이미지 타입을 출력할 준비를 할 것이다.

이렇게 각 미디어들에 대한 표현정보가 세팅이 되면, 이 정보들을 바탕으로 선택된 미디어들에 대한 미



디어 플레이어 객체들을 생성한다(S142610). 각 미디어 플레이어(14220~14228)는 생성될 때 미디어 큐(1424)를 하나씩 퍼블릭 타입으로 할당받는데, 자바의 동기화 기능을 이용하여 임계 섹션(critical section)을 구성한다.

선택된 미디어 플레이어들이 생성이 되고 나면, 클라이언트 자바 애플릿(142)은 카타로그 표현을 시작하기 위한 초기화 입력 데이터를 구성하여(S142612), 자바 이벤트 핸들러에게 넘겨준다. 또한 재생 도중에 입력되는 동적인 사용자 입력값들은 단계(S14602)의 판단에 따라 클라이언트 자바 애플릿(142)을 통해 바로 자바 이벤트 핸들러로 입력되어 파싱된다(S142604).

이때 파싱된 입력값은 순차적으로 속도조절(S142614), 임의의 재생 위치값 변경(S142616), 연주(PLAY)(S142618), 일시정지(PAUSE)(S142620), 정지(STOP)(S142622), 빠르게 감기/되감기(FF/REW)(S142624), 보임/숨김(SHOW/HIDE)(S142626), 초당 프레임율(FPS) 설정(S142628) 인자가 판단되고, 해당 경우에 대한 세부 스텝을 수행하게 된다.

만약 속도조절에 관한 이벤트가 발생되었다면, 입력된 더블값으로 속도값을 세팅한다(S142638). 속도값의 단위는 1.0을 기준으로 소수점 첫째자리까지 구성된다. 반대방향으로 재생할 때에는 (-)값을 입력한다. 이렇게 입력된 속도값은 다시 서버 시스템(SS1)으로 전달되어 서버 시스템이 해당 속도로 수행되도록 한다(S142640).

임의의 재생위치로 이동하고자 할 경우에는 재생위치 값을 입력된 값으로 세팅한다(S142642). 재생위치값의 단위는 초 단위로 구성한다. 재생위치에 대한 내부 값을 변경 한 후에는 클라이언트 자바 애플릿(142)의 디스플레이 바(display bar)를 물리적으로 이동시켜 사용자가 추적할 수 있도록 한다(S142644). 이 경우 디스플레이 바는 생략할 수 도 있다. 이와 동시에 서버 시스템(140)에 변경된 위치값을 전달해 거기에서부터 데이터가 읽혀지도록 한다(S142646).

연주(PLAY)에 대한 입력값이 전달되면 속도값을 정상속도인 1.0으로 세팅하며(S142632), 정지(STOP)인 경우에는 속도값을 0.0으로 세팅하고(S142622), FF나 REW인 경우에는 속도값을 각각 3.0과 -3.0으로 세팅한다. 일시정지(PAUSE)인 경우에는 토글(toggle) 기능으로 속도값을 1.0과 0.0을 반복하여 세팅한다. 이와 동시에 서버 시스템(SS1)에 변경된 위치값을 전달해 거기에서부터 데이터가 읽혀지도록 한다(S142646).

사용자는 또한 동기화 엔진(1426)을 통해 특정 미디어 플레이어를 보였다(SHOW)(S142648), 숨겼다(HIDE)(S142650) 할 수 가 있다. 즉, 재생 중에 어느 때라도 자신이 원하는 미디어만 동적으로 표현할 수 가 있다. 이때, 숨은(HIDE) 미디어 플레이어들은 다시 보임(SHOW)으로 될 때까지 더 이상 서버 시스템(SS1)로부터 데이터를 받지 않기 때문에 네트워크 대역폭을 효율적으로 활용할 수가 있다. 또한, 상기 단계(S142628) 판단결과, 초당 프레임율(FPS)이 변경설정된 경우 비디오 서버(1400)에 바뀐 FPS값을 전달한다(S142652).

사용자로부터의 입력 사항을 모두 처리한 동기화 엔진(1426)은 지속적으로 각 플레이어들의 동기화를 관리하고 유지시키기 위해 자신의 객체 레퍼런스를 각 미디어 플레이어들에게 공개(open)한다. 이 레퍼런스를 통해 각 미디어 플레이어들은 현재의 동기화 상황에 대한 동기화 변수 및 플래그들을 참조할 수 가 있다.

도 5는 미디어 플레이어의 동작을 설명하는 흐름도이다. 미디어 플레이어(1422)는 클라이언트 자바 애플릿(142)을 jar 파일로 구성해 웹서버(12)에 탑재할 때 프로그램 사이즈를 줄이고 다운로드 시간을 단축하기 위해서 해당 어플리케이션에 필요한 미디어 객체들만 구성(configurable)되어질 수 도 있다.

예를들어, 파노라마 기능만을 이용해 카타로그를 구성하고 싶을 경우에는 이 단계에서 이미지 플레이어(14224) 객체 클래스만 클라이언트 jar 파일에 탑재시킨다. 만약 이미지와 텍스트를 이용해서 슬라이드 쇼를 구성하고 싶다면 이미지 플레이어 클래스와 텍스트 플레이어 클래스만 탑재시킨다. 만약 모든 기능을 기본적으로 내장하고 사용자의 동적 요구에 따라 특정 미디어 클래스 객체를 선택하고자 한다면 모든 클래스들을 jar 파일에 탑재시킨다.

도 3에 도시된 미디어 플레이어는 기본적으로 자바의 멀티 쓰레드(Multi Thread) 기능을 이용하기 때문에, 하나 이상의 선택된 미디어 플레이어들이 동시에 병렬적으로 수행이 될 수 있다. 즉, 비디오 플레이어 객체 생성단계(S142210), 오디오 플레이어 객체 생성단계(S142226), 이미지 플레이어 객체 생성단계(S142242), 텍스트 플레이어 객체 생성단계(S142266), HTML 플레이어 객체 생성단계(S142284)가 동시에 병렬적으로 수행이 된다. 이렇게 함으로써 한 가지 이상의 미디어가 동시에 표현이 될 수 있는 것이다.

도 5를 참고하여 미디어 플레이어에 대한 제어 흐름도를 자세히 설명하면 다음과 같다. 우선 사용자가 비디오 미디어의 출력을 선택했는지를 판단한다(S142200). 만약 선택하였다면 비디오 플레이어 객체를 퍼블릭 타입(public type)으로 하나 생성한다(S142210). 이 생성단계에서 비디오 플레이어(14220)는 이미 오픈된 동기화 엔진(1426) 객체의 레퍼런스를 참조 받음으로써 동기화 엔진(1426)의 최신 동기화 정보들을 지속적으로 얻을 수 있게 된다. 비디오 플레이어(14220) 객체는 이어서 실제 미디어 디스플레이 엔진(1428)이 출력할 수 있는 픽셀 이미지로 변환하기 위한 비디오 디코더 객체를 생성한다(S142212). 이 비디오 디코더 객체는 독자적으로 수행될 수 있는 쓰레드(Thread) 객체로 생성된다.

생성된 비디오 디코더 객체는 미디어 큐(1424)에서 현재 시각에 재생할 비디오 프레임이 있는 지를 체크한다(S142214). 이 판단은 동기화 엔진(1426)의 재생 속도와 재생 위치, 그리고 실제 비디오 프레임을 내의 논리적 시작 시간과 논리적 끝 시간을 비교하여 이루어진다. 이때 만약 동기화 엔진(1426)의 속도가 0라면 정지 상태로 판단하여 초(

(1 초)만큼 자신의 쓰레드를 슬립(sleep)한 후(S142216), 다시 단계(S142214)로 가서 지속적인 동기화 상황 체크를 반복한다. 만약 속도가 0 이 아니면서도 현재의 논리적 시각내에 해당하는 비디오 프레임이 없을 경우에도 단계(S142216)를 진행한다.



만약 미디어 큐(1424) 안에 현재 논리적 시각에 해당하는 비디오 프레임이 있다면, 그 프레임을 미디어 큐(1424)에서 가져 온 후(S142218), MPEG-1 디코딩을 수행한다(S142220). 현재 본 비디오 플레이어에서 재생 가능한 비디오 프레임들은 MPEG-1 비디오 스트림을 기반으로 구성되어 있다. 디코딩이 완료된 후에는 디스플레이 엔진(1428)의 비디오 디바이스에 의해 화면에 출력되게 되고(S142222), 이 출력이 현재 프레임의 재생시간 - 초당 프레임률 - 동안 화면상에 그대로 표시될 수 있도록 비디오 디코더 쓰레드를 남은 프레임 재생시간 만큼 슬립(sleep)시킨 후(S142224), 다음 프레임의 재생을 위해 다시 단계(S142214)로 올라간다. 이때 한 번 출력된 비디오 프레임들은 자동적으로 미디어 큐(1424)에서 제거되어 메모리 부하를 줄여 준다.

한편, 단계(S142202)에서 오디오 미디어가 사용자로부터 선택되었는지가 단계(S142210)와 거의 동시에 판단된다. 만약 오디오 미디어를 사용자가 선택하였다면 단계(S142210)부터 단계(S142224)와 거의 같은 절차를 단계(S142226)에서부터 단계(S142240)에서 수행한다. 다만, 차이점이 있다면, 오디오 디코딩의 기본 타입이 초당 8000Hz의 샘플링률을 가진 선 마이크로시스템즈사의 Au 타입과 초당 13000 bps로 압축되어 전송되는 GSM(The Global System for Mobile telecommunication) 타입을 지원한다는 것이다. 초당 13 kbps로 전송되는 GSM 오디오 타입을 지원함으로써 28K 모뎀에서도 실시간으로 오디오 지원이 가능하다.

한편, 단계(S142204)에서는 이미지 미디어가 사용자로부터 선택되었는지를 판단한다. 만약 선택되었다면 이미지 플레이어(14224) 객체를 생성하고 동기화 엔진(1426)의 객체 레퍼런스를 참조하는데(S142242), 이미지 디코더 객체를 생성함에 있어 2D 이미지와 3D 이미지 모델링을 미디어 적용 타입에 따라 선별적으로 취급할 수 있는 서버 디코딩 컴포넌트 객체들을 함께 생성한다(S142244, S142246).

비디오 플레이어(14220)와 오디오 플레이어(14222)가 연속적인 미디어 형태를 다루는데 반해, 이미지 플레이어(14224), 텍스트 플레이어(14226), HTML 로더(14228)는 비연속적인 미디어 형태를 다루기 때문에 도 3에서 언급한 멀티미디어 템플레이트 파일 시스템(1402)의 해당 템플레이트 파일을 단계(S142250, S142274, S142290)에서 서버 시스템(S1)으로부터 읽어와 사용한다.

비디오를 제외한 나머지 비연속적 미디어의 멀티미디어 템플레이트 파일(140208, 140210, 140212)은 상기한 바와같이 다음과 같은 형식으로 구성되어 진다: {미디어를 가져올 미디어 파일 디렉토리, 미디어 타입, 논리적 시작 시간, 논리적 끝 시간}.

여기서 미디어 타입은 파일의 확장자로 인식한다. 예를 들어, gif/jpeg등은 이미지 타입, txt는 텍스트, html/htm은 HTML 문서 타입을 나타낸다. 본 템플레이트 파일은 사용자가 텍스트 기반의 에디터로 작성할 수 있다. 기존의 방식과 다르게 모든 미디어들의 내용을 하나의 이진 파일에 통합해 놓지 않았기 때문에 언제나라도 변경 사항의 수정이 용이하다.

이 정보를 바탕으로 단계(S142252, S142274, S142294)에서 실제의 미디어 데이터를 서버 시스템(SS1)으로부터 실시간으로 읽어와 논리적 재생 시간을 결합한 완전한 미디어 프레임을 완성한다. 이렇게 완성된 미디어 프레임 조각들은 해당 디코딩 단계(S142260, S142258)를 거쳐 디코딩되고, 디스플레이 엔진(1428)에 표현된다(S142262, S142280, S142296).

화면에 표시된 후에는 각자의 재생 시간 동안 재생 시간을 보장하기 위해 각 디코더 객체 쓰레드가 슬립하게 되는데, 연속적 미디어와는 다르게 자신의 재생시간이 끝나면 바로 화면에서 제거시켜 미디어의 불연속성을 보장시킨다(S142264, S142282, S142298). 아울러 비연속성의 보장을 통하여 네트워크의 대역폭을 효과적으로 활용할 수 있다.

여기에서 주의 깊게 살펴보아야 할 부분은 이미지 디코더에서 3D 이미지 모델링 판단단계(S142258)이다. 이 부분은 크게 모션 모듈과 파노라마 모듈로 나뉘어 진다. 모션 모듈은 서로 다른 각도에서 촬영된 18장 이상의 연속적인 2D 이미지들을 순차적으로 받아들이면서 3차원적으로 모델링해주는 기법으로 마치 3D 객체를 보는 듯한 효과를 주게된다. 여기에서 패닝(panning)기능을 제공하여 이미지들이 연속적으로 바뀌게 함으로써 마치 실제 3차원 물체를 둘러보는 듯한 느낌을 주게 할 수 있다.

이러한 기능은 웹 카탈로그 시스템에서 전시된 상품을 실제 3차원 모양 그대로 모델링해주는 효과를 나타낸다. 다음으로 파노라마 모듈은 한장의 2D 이미지를 가지고 3차원적인 360도 패닝 효과를 구현해주며, 아울러 이미지 내의 특정 공간을 클릭했을 때 다른 미디어 플레이어와 연동시켜주는 기능을 포함한 다.

도 6은 서버 시스템을 동작을 설명하는 흐름도로서, 서버 시스템(SS1)이 웹서버(12)로부터 비디오 서버 서블릿(servlets)을 구동하는 단계(S140000)에서부터 시작한다. 다음으로 멀티미디어 템플레이트 파일 시스템(1402)으로부터 비디오 템플레이트 파일(140206)을 메모리에 로드시킨 후 각각을 객체화시켜 해당 미디어 플레이에 대한 사용자 요구(request)에 반응하도록 한다(S140002). 이렇게 메모리에 템플레이트 파일(140206)을 로드시켜 놓는 이유는 사용자 반응시간을 최대한으로 줄이고 서비스 성능을 개선시키기 위함이다. 템플레이트 파일의 로드시에 실제 콘텐츠 데이터는 하드 디스크 상에 존재하며, 물리적 저장 위치의 정보와 각 프레임당 시간 정보를 담고 있는 헤더 부분들만 메모리에 탑재된다.

비디오를 제외한 나머지 미디어 즉, 오디오, 이미지, 텍스트, HTML 데이터들은 사용자 요구가 있을 때(S140038), 해당 파일을 디스크로부터 읽어 들여(S140040), 클라이언트 단말기(C1)와의 스트리밍 프로토콜에 따라 클라이언트 단말기(C1)로 데이터를 전송한 후(S140042), 세션(Session)을 차단한다(S140044).

비디오 데이터의 경우는 비디오 서버 엔진(1400)에 의해 특별하게 취급된다. 다른 미디어에 비해 상대적으로 많은 데이터량과 많은 I/O 자원을 소모하고 동적으로 변하는 네트워크의 대역폭 상황을 탄력적으로 반영하여 스트리밍하여야 하기 때문이다. 일단 사용자로부터 비디오 서블릿(Video Servlet)에 대한 세션 접속이 일어나면(S140004), 다음과 같은 명령어들이 URL(Uniform Resource Locator) 형태로 클라이언트 단말기(C1)에서 비디오 서블릿으로 전달되게 된다:

http://videoserver:port/VideoServlet?file=title1.mpgc&fps=10&its=0&speed=1

이렇게 입력된 명령어안에서 비디오 서버 엔진(1400)은 일단 비디오 타이틀명을 파싱한 후(S140006), 파싱된 타이틀에 대한 메모리의 해당 템플레이트 객체를 선택한다(S140008). 다음으로 실제 비디오 데이터를 읽어 들일 재생위치 값을 입력된 파라미터로부터 결정하는데(S140010), 이것이 0값이면 비디오 템플레이트 객체에서 처음의 오프셋 포인터(offset pointer)로 이동(S140012)하고, 아니면 비디오 템플레이트 객체에서 해당 위치의 오프셋 포인터로 이동한다(S140014).

다음으로 재생속도를 판단하는데(S140016), 만약 변경되었으면 재생속도를 변경하여 반영하고(S140020), 아니면 기존의 속도를 그대로 유지시킨다(S140018).

이어서, 프레임률(FPS; Frames per Second)의 변경을 판단하는데(S140022), 현재의 네트워크 대역폭의 변동 상황을 반영한 사용자로부터의 FPS 입력값이 기존의 값과 차이가 나면 새로 입력된 값으로 FPS 값을 설정하고(S140024), 아니면 기존의 FPS를 그대로 유지한다(S140026).

이상과 같이 모든 입력 파라미터에 대한 파싱 절차가 끝나면, 실제 현재의 오프셋(offset)에서부터 현재의 속도로 비디오 스트림 파일을 읽어 들인다(S140028). 비디오 데이터의 판독은 1초 단위로 반복되는 데, 1초 분량의 비디오 데이터가 읽히는 대로 현재의 FPS 만큼만 스트림을 패킷화하여(S140030), 클라이언트 단말기(C1)에게 패킷을 전송한다(S140032). 이 작업은 EOF(End of File)를 만나거나(S140034), 아니면 세션이 강제적으로 끝났을 때(S140036) 까지 반복된다.

도 9는 미디어 수신 엔진의 동작을 설명하는 흐름도로서, 미디어 수신 엔진(1420)은 클라이언트 자바 애플릿(142)이 구동될 때 미디어 수신 엔진 객체를 생성(S142000)함으로써 구현된다. 일단 미디어 수신 엔진(1420) 객체가 생성되고 나면 현재의 네트워크 대역폭을 계산하기 위해 미리 설정한  $\alpha$  초만큼의 패킷을 사전에 미리 서버 시스템(SS1)에게 요구한다. 이때 비디오의 타이틀명, FPS, 재생속도=1, 재생위치=0과 같이 초기 파라미터를 서버 시스템(SS1)에 전달한다(S142002). 이어서 미디어 수신 엔진(1420)은 서버 시스템(SS1)으로부터  $\alpha$  초만큼의 패킷을 전송 받은 후 초당 평균 수신시간(L)을 계산한다(S142004).

그후 평균 수신 시간값(L)으로 현재의 네트워크 상태에서의 최적의 FPS를 계산한다(S142006). 이 부분에서 본 발명 시스템은 종래의 30 FPS 기반의 MPEG 스트림을 현재 대역폭 상황에 따라 선택적으로 수신 받을 수 있는 메커니즘을 제공함에 따라 종래의 정상 MPEG 스트림이 할 수 없었던 저대역폭, 동적 대역폭 상황에 탄력적으로 서비스를 할 수 있게 된다. 일단 현재 상태에서의 최적의 FPS가 계산이 되면, 실제 서버 시스템(SS1)으로부터 데이터가 전송될 때까지 비동기적인 리스(listen)상태에 들어가게 된다. 따라서, 본 발명 시스템에서는 실제 서버 시스템(SS1)으로부터 데이터가 올 때에만 깨어나기(wakeup) 때문에 다른 프로세스들이 작업을 진행할 수 있는 기회를 제공한다(S142008).

서버 시스템(SS1)으로부터 데이터 패킷이 도착하면(S142010), 1초당의 데이터 패킷을 수신한다(S142012). 다음으로 전송 받은 패킷의 헤더 부분에서 미디어 타입을 파싱한다(S142014). 그리고 해당 미디어 쿼(1424)에 데이터 패킷을 삽입한다(S142016).

이러한 작업을 계속 반복하여 연속되는 패킷들을 미디어 쿼(1424)에 삽입하게 되는데, 수신도중에 갑자기 대역폭상황이 나빠져 마지막으로 전송 받은 패킷의 논리적 끝 시간이 현재 재생중인 논리적 시간(재생위치)을 초과하는 경우가 발생할 수 있다(S142018). 이러한 경우에는 어느 정도의 시간만큼 버퍼링하는 단계가 필요하게 된다. 버퍼링 작업은 동기화 엔진(1426)의 재생속도를 0으로 세팅하고 잠시 블록상태로 들어가는 것으로부터 시작되게 된다(S142020).

다음으로 초당 평균 수신값(L)을 기준으로 최적의 버퍼링 시간을 예측한다(S142022). 그리고 버퍼링 시간 만큼 데이터 패킷을 수신한 후 미디어 쿼(1424)에 삽입한다(S142024). 버퍼링 작업이 완료되면 다시 동기화 재생속도를 1로 세팅함으로써 멈췄던 지점으로부터 재생을 다시 시작한다(S142026). 초당 평균 수신시간(L)은 매 5초마다 재 계산하게 되고(S142028), 미디어 수신 엔진(1420)은 한 패킷의 수신 작업이 끝날 때마다 리스(listen)상태에서 다음 패킷을 기다리게 된다(S142008).

도 10은 미디어 디스플레이 엔진의 블록도를 나타낸다. 미디어 디스플레이 엔진(1428)은 세개의 API로 이루어져 있는데 GUI를 담당하는 자바 AWT(Abstract Windows Toolkit) API(Application Program Interface)(42802), 오디오 출력을 담당하는 자바 오디오 API(42806), 그리고 HTML의 출력을 담당해주는 웹 브라우저 API(42820)로 나누어진다. 자바 AWT API(42802)는 비디오 디스플레이 디바이스(42804)와 연결되어 시각적인 표현을 담당하며, 자바 오디오 API(42806)는 오디오 출력 디바이스(42808)와 연결되어 오디오의 출력을 담당해준다. 또한, 웹 브라우저 API(42810)는 HTML 출력에 관한 작업을 수행한다.

도 11은 도 2에 도시된 멀티미디어 H/W장치(8,9)의 상세 블록도이다. 멀티미디어 H/W 장치(8,9)는 사운드 카드(1602)와 VGA OUT(1606)으로 구성되어 있다. 사운드 카드(1602)는 스피커 시스템(1604)과 연결되어 실제 오디오 사운드를 아날로그 파형으로 변환하여 출력하게 된다. VGA OUT(1606)은 사용자의 선택에 따라 화면 시그널을 모니터(1608)나, LCD(1610), 또는 빔 프로젝터(1612)로 출력하게 된다.

#### 발명의 효과

상기한 바와같이 본 발명에서는 정적인 HTML 문서내에 멀티미디어 웹 카타로그에 필요한 기능들을 동적으로 실시간상에 구현하기 위한 자바 애플릿을 클라이언트 애플리케이션으로 삽입하고, 서버 시스템에 현재 표현에 필요한 각종 멀티미디어 데이터를 각각의 특성에 맞게 동적으로 요청하여 실시간적으로 웹 서버를 통해 웹 브라우저로 스트리밍함으로써 HTML상에 함께 혼합하여 카타로그 내용들을 웹 브라우저로 효과적으로 표현할 수 있다.

또한, 본 발명에서는 HTML 문서내에 삽입한 자바 애플릿을 클라이언트 프로그램으로 구동하고, 그것의 동기화 엔진과 스트리밍 기능 등을 적절히 혼합하여 실시간상에 선택된 비디오, 오디오, 2D/3D 이미지, 애니메이션, 텍스트, 애니메이션, HTML 등의 미디어 프레임들을 웹서버와 상기 비디오 서버 엔진을 통해

논리적으로 송/수신 결합시켜 줌으로써, 종래의 HTML이나 DHTML(Dynamic-HTML) 문서, 플러그-인(Plug-In) 프로그램(S/W), 자바 스크립트로도 표현하기 어려웠던 동적 멀티미디어 웹 카타로그 정보들을 웹 브라우저 내에 효과적으로 표현해 줄 수 있게 되었다.

더욱이, 본 발명 시스템은 하드웨어에 독립적인 순수 자바(Pure-Java) 기반으로 설계되어 자바 버추얼 머신(Java Virtual Machine)을 탑재한 PC(개인용 컴퓨터)의 웹 브라우저에서는 물론, 퍼스널 자바(PersonalJava)를 포팅(porting)한 일반 휴대용 통신단말기(PCS, PHS), 휴대 정보 단말기(PDA), 셋톱 박스(settop box), 디지털 텔레비전(Digital TV), 웹폰(Web Phone) 및 자바 오퍼레이팅 시스템(JavaOs)을 탑재한 자바 칩(Java Chip) 등에서 실행이 가능하다.

또한, 스트리밍 기반으로 이루어져 있기 때문에 인터넷의 대역폭에 따라 필요한 미디어들만 선택적으로 받아 볼 수 있고, 아울러 프레임 기반의 가상 VCR을 이용한 지능적 재생이 가능하며, 종래의 비디오, 오디오 등의 연속적 미디어 스트리밍 방법에서 이미지, 텍스트, HTML 등의 비연속적 미디어 스트리밍 방식을 새롭게 고안하여 웹상에서의 가장 적합한 멀티미디어 표현 방식을 제공할 수 있다.

이상에서는 본 발명을 특정의 바람직한 실시예를 예를 들어 도시하고 설명하였으나, 본 발명은 상기한 실시예에 한정되지 아니하며 본 발명의 정신을 벗어나지 않는 범위내에서 당해 발명이 속하는 기술분야에서 통상의 지식을 가진자에 의해 다양한 변경과 수정이 가능할 것이다.

### (57) 청구의 범위

#### 청구항 1

웹 브라우저와 자바 버추얼 머신을 포함하는 클라이언트 단말기와 웹서버를 포함하는 서버 시스템이 인터넷망을 통하여 연결된 클라이언트/서버 시스템 환경에서 자바를 이용한 동적 멀티미디어 웹 카타로그 시스템에 있어서,

비디오, 오디오, 2D/3D 이미지, 애니메이션, 텍스트 및 HTML을 포함하는 멀티미디어 데이터를 스트리밍에 적합하도록 제반정보를 포함한 템플레이트 파일로 이루어진 멀티미디어 템플레이트 파일시스템과;

사용자가 웹 브라우저를 통해 접속할 때 클라이언트 자바 애플릿 정보가 포함된 HTML 카타로그 문서를 웹 브라우저로 제공하며, 상기 클라이언트 자바 애플릿 정보에 따라 웹 브라우저에 생성된 클라이언트 자바 애플릿의 요구가 있는 경우 오디오, 2D/3D 이미지, 애니메이션, 텍스트 및 HTML을 포함하는 멀티미디어 데이터에 대하여는 직접 멀티미디어 템플레이트 파일시스템을 액세스하여 클라이언트 자바 애플릿으로 스트리밍 해주는 웹서버와;

상기 클라이언트 단말기로부터의 비디오 데이터에 대한 요구에 따라 그 내용을 해석하여 웹서버를 통하여 클라이언트 자바 애플릿으로 비디오 데이터를 실시간 상으로 스트리밍해주기 위한 비디오 서버 엔진과;

상기 웹서버에 접속할 때 HTML 카타로그 문서 내에 삽입되어 웹 브라우저로 다운로드되어 자바 버추얼 머신에 의해 생성되며 멀티미디어 웹 카타로그에 필요한 기능들을 동적으로 실시간 상에 구현할 수 있는 클라이언트 자바 애플릿으로 구성되며,

상기 클라이언트 자바 애플릿은

클라이언트의 웹 브라우저 상에서 해당 논리적 시각에 필요한 비디오를 포함한 멀티미디어 데이터를 서버 시스템에 요청하고, 웹서버를 통해 전달받은 멀티미디어 데이터를 동적으로 결합하여 재생시키기 위한 동기화 엔진과,

상기 웹서버와 비디오 서버 엔진으로부터 카타로그 멀티미디어 데이터 패킷을 HTTP를 통하여 스트리밍 방식으로 전달받아 미디어큐에 임시로 저장하는 미디어 수신 엔진과,

상기 동기화 엔진의 시간정보, 재생정보, 속도정보 등의 가상 VCR 정보를 계속 체크하면서 해당되는 논리적 시각에 해당 미디어 데이터를 미디어큐로부터 읽어들이 각 파일 포맷에 맞게 디코딩하고 해당 장치 디바이스로 재생되도록 제어하는 미디어 플레이어 엔진과,

상기 미디어 플레이어 엔진에서 디코딩된 미디어 데이터를 재생하기 위한 디스플레이 엔진으로 구성되는 것을 특징으로 하는 자바를 이용한 동적 멀티미디어 웹 카타로그 시스템.

#### 청구항 2

제1항에 있어서, 상기 동기화 엔진은 상기 HTML내에 삽입된 자바 애플릿으로부터 멀티미디어 데이터의 혼합에 관한 정보를 전달받아 선택된 미디어들에 대하여 웹서버로부터 오는 각각을 논리적 시간상에 동기화하여 각각의 표현 시간을 관리해 주는 수명관리수단과,

웹 카타로그에 관련된 제반정보를 담고있는 템플레이트 파일을 실시간적으로 해석하여 상기 자바 애플릿과 일 이상의 HTML 프레임 내에 그 객체들을 동적으로 표현하고 관리하는 서비스 매니저수단으로 구성되는 것을 특징으로 하는 자바를 이용한 동적 멀티미디어 웹 카타로그 시스템.

#### 청구항 3

제1항에 있어서, 상기 웹서버로부터 전달되는 각 미디어 데이터들은 모두 프레임 단위로 존재하며, 각각의 프레임은 실제 표현되어질 이진 압축 데이터, 논리적 표현시작시간 및 논리적 표현끝시간으로 구성되고,

상기 제반정보는 웹서버 내의 저장위치, 매체타입, 매체이름, 표현시작시간 및 표현끝시간으로 구성되는 것을 특징으로 하는 자바를 이용한 동적 멀티미디어 웹 카타로그 시스템.

**청구항 4**

제1항에 있어서, 상기 멀티미디어 템플레이트 파일 시스템은 실제 미디어 데이터를 핸들링할 메타 데이터(metadata)로 이루어진 템플레이트 파일 디렉토리, 실제 미디어 데이터들을 저장하고 있는 물리적인 파일 시스템인 미디어 파일 디렉토리로 나누어지며,

상기 미디어 파일 디렉토리는 비디오 파일, 오디오 파일, 이미지 파일, 텍스트 파일 및 HTML 파일로 구성되고,

상기 템플레이트 파일 디렉토리는 클라이언트 단말기로부터의 요구(request)가 도착했을 때 빠르게 스트리밍을 시작할 수 있도록 서버 시스템이 초기화 될 때 서버 시스템의 메모리로 자동 로드되며 물리적으로 비디오 파일의 헤더 부분에 위치된 비디오 템플레이트 파일, 이미지 템플레이트 파일, 텍스트 템플레이트 파일 및 HTML 템플레이트 파일로 구성되고,

상기 템플레이트 파일은 미디어를 가져올 미디어 파일 디렉토리, 미디어 타입, 논리적 시작 시간 및 논리적 끝 시간으로 구성되어, 상기 클라이언트 자바 애플릿이 처음 초기화될 때 동기화 엔진이 참조하여 기본 동기화 전략을 수립하는데 사용되며, 사용자가 텍스트 기반의 에디터로 작성 가능한 것을 특징으로 하는 자바를 이용한 동적 멀티미디어 웹 카타로깅 시스템.

**청구항 5**

제1항에 있어서, 상기 각 해당 논리적 시간 내에 재생되는 미디어들은 비디오, 오디오, 텍스트, 2D 스틸 이미지, 3D 모델링을 위한 2D 이미지 세트, 애니메이션 이미지, HTML 문서를 포함하며,

상기 미디어들 각각은 클라이언트 자바 애플릿 내 및 다른 웹 브라우저 창에서 서로의 논리적 재생 시간에 독립적으로 수행될 수 있고, 3D 모델링을 위한 2D 이미지 세트는 이미지 플레이어에서 한편의 3D 모델링 이미지 및 애니메이션을 만들기 위한 독립적인 형태로 이용되며, 3D 모델링을 위한 2D 이미지 세트를 제외한 다른 모든 미디어들은 독립적인 재생시간 단위로 쪼개어진 프레임들로 구성되는 것을 특징으로 하는 자바를 이용한 동적 멀티미디어 웹 카타로깅 시스템.

**청구항 6**

제5항에 있어서, 상기 웹 카타로깅을 동적으로 수행할 때 각각의 미디어는 독자적 재생, 하나 이상의 동일 및/또는 서로 상이한 미디어 조합 중 어느 하나의 형태로 미디어가 재생되며,

상기 재생되는 미디어 조합은 비디오와 텍스트 자막으로 이루어진 제1미디어 조합, 비디오와 오디오로 이루어진 제2미디어 조합, 비디오, 오디오 및 텍스트 자막으로 이루어진 제3미디어 조합, 비디오, 오디오, 텍스트 자막 및 스틸 이미지로 이루어진 제4미디어 조합, 비디오, 오디오, 제1언어 텍스트 자막 및 제2언어 텍스트 자막으로 이루어진 제5미디어 조합, 비디오, 오디오, 제1언어 텍스트 자막, 제2언어 텍스트 자막 및 제3언어 텍스트 자막으로 이루어진 제6미디어 조합, 비디오, 오디오 및 HTML 문서로 이루어진 제7미디어 조합, 비디오, 오디오, 텍스트 자막 및 HTML 문서로 이루어진 제8미디어 조합, 비디오, 오디오, 텍스트 자막, HTML 문서 및 스틸 이미지로 이루어진 제9미디어 조합, 스틸 이미지 및 텍스트 자막으로 이루어져서 슬라이드 쇼를 구현하는 제10미디어 조합, 스틸 이미지, 텍스트 자막 및 각 이미지 당 오디오로 이루어져서 오디오 슬라이드 쇼를 구현하는 제11미디어 조합, 애니메이션 이미지에 의해 롤링 배너광고를 구현하는 제12미디어 조합, 애니메이션 이미지 및 각 이미지 당 오디오로 이루어져서 사운드 롤링 배너 광고를 구현하는 제13미디어 조합, 3D 파노라마 이미지의 제14미디어 조합, 다른 각도에서 찍은 2D 이미지의 연속에 의해 3D 사진 모델링 효과를 구현하는 제15미디어 조합 중 어느 하나의 조합인 것을 특징으로 하는 자바를 이용한 동적 멀티미디어 웹 카타로깅 시스템.

**청구항 7**

제1항에 있어서, 상기 미디어 플레이어 엔진은 웹 브라우저를 통하여 입력한 사용자 입력 파라미터를 파싱하여 사용자가 웹 카타로그 표현을 위해 선택한 미디어들에 대한 플래그 설정에 따라 생성된 적어도 일 이상의 미디어 플레이어로 구성되고,

상기 미디어 플레이어는 비디오 플레이어, 오디오 플레이어, 이미지 플레이어, 텍스트 플레이어 및 HTML 로더로 구성되며, 이들은 동시에 생성되어 일 이상의 미디어가 동시에 표현되는 것을 특징으로 하는 자바를 이용한 동적 멀티미디어 웹 카타로깅 시스템.

**청구항 8**

제1항에 있어서, 상기 미디어 디스플레이 엔진은 비디오 디스플레이 디바이스와 연결되어 시각적인 표현을 담당하는 자바 AWT API, 오디오 출력 디바이스와 연결되어 오디오 출력을 담당하는 자바 오디오 API 및 HTML의 출력을 담당하는 웹 브라우저 API로 구성되는 것을 특징으로 하는 자바를 이용한 동적 멀티미디어 웹 카타로깅 시스템.

**청구항 9**

제1항에 있어서, 서버 시스템의 성능 및 네트워크의 부하를 고려하여 특정 미디어 데이터가 분산 배치되며 미디어 파일 디렉토리와 TCP/IP를 기반으로 하여 상호 물리적인 저장위치가 공유되는 적어도 일 이상의 콘텐츠 서버를 더 포함하는 것을 특징으로 하는 자바를 이용한 동적 멀티미디어 웹 카타로깅 시스템.

**청구항 10**

브라우저와 자바 부추일 머신을 포함하는 클라이언트 단말기와, 웹서버를 포함하는 서버 시스템이 인터넷망을 통하여 연결된 클라이언트/서버 시스템 환경에서 자바를 이용한 동적 멀티미디어 웹 카타로깅 방법에 있어서,

상기 웹 브라우저를 사용하여 서버 시스템의 웹서버에 접속하는 단계와,

상기 웹서버로부터 클라이언트 자바 애플릿 정보가 포함된 HTML 카타로그 문서를 자신의 웹 브라우저로 다운로드 받는 단계와,

상기 웹 브라우저에서 HTML 카타로그 문서의 HTML 태그를 파싱한 후 자바 버추얼 머신을 통해 멀티미디어 웹 카타로그에 필요한 비디오, 오디오, 이미지, 텍스트, HTML 데이터 중 적어도 하나 이상을 포함하는 멀티미디어 데이터를 서버 시스템에 요청하고, 웹서버를 통해 전달받은 멀티미디어 데이터들을 동적으로 결합하여 재생시키기 위한 동기화 엔진과 서버 시스템으로부터 웹 카타로그 미디어의 데이터 패킷을 스트리밍 방식으로 전달받아 미디어큐에 보관하는 미디어 수신 엔진을 포함하는 클라이언트 자바 애플릿 객체를 생성하고 정해진 HTML 문서 내에 GUI를 생성하는 단계와,

그후 사용자의 초기 입력값이 웹 브라우저를 통하여 입력되면 사용자 입력 파라미터를 파싱하여 사용자가 원하는 카타로그 표현을 위해 선택한 미디어들에 대한 플래그를 설정하는 단계와,

상기 각 미디어에 대한 표현정보가 설정되면 설정된 정보에 기초하여 선택된 미디어들에 대응하여 각각 수신받은 각각의 미디어 데이터들을 각 파일 포맷에 맞게 디코딩하고 해당 장치 디바이스로 재생해주는 비디오 플레이어, 오디오 플레이어, 이미지 플레이어, 텍스트 플레이어 및 HTML 플레이어를 포함하는 미디어 플레이어 객체를 생성함과 동시에 각각의 미디어 플레이어 객체에 대응하는 미디어큐를 하나씩 퍼블릭 타임으로 할당받는 단계와,

상기 미디어 플레이어가 생성된 경우 클라이언트 자바 애플릿은 카타로그 표현을 시작하기 위한 초기화 입력 데이터를 구성하여 자바 이벤트 핸들러에 넘겨주는 단계와,

상기 파싱된 입력값에 기초하여 속도조절, 재생위치 및 재생동작 선택에 따른 설정된 속도값과, 설정된 재생위치와, 설정된 초당 프레임율(FPS)을 동기화 엔진으로부터 웹서버를 통해 서버 시스템으로 전달하는 단계와,

상기 요구에 따라 비디오 데이터에 대한 요구는 비디오 서버 엔진이 그 내용을 해석하여 웹서버를 통해 클라이언트 자바 애플릿으로 스트리밍해주며, 비디오 데이터 이외의 오디오, 이미지, 텍스트, HTML 데이터들은 웹서버가 직접 멀티미디어 템플레이트 파일 시스템으로부터 읽어 들여 클라이언트 자바 애플릿으로 스트리밍해주는 단계와,

상기 미디어 수신 엔진에 의해 카타로그 미디어 데이터 패킷을 스트리밍 방식으로 수신하여 미디어큐에 보관하는 단계와,

수신받은 각각의 미디어 데이터들을 동기화 엔진의 재생시간 제어에 따라 동기화가 이루어지면서 미디어 플레이어에 의해 각 파일 포맷에 맞게 디코딩하여 해당하는 장치 디바이스로 재생해주는 단계와,

상기 미디어 플레이어에서 디코딩된 미디어 데이터를 디스플레이 엔진에 의해 재생하는 단계로 구성되는 것을 특징으로 하는 자바를 이용한 동적 멀티미디어 웹 카타로그 방법.

#### 청구항 11

제10항에 있어서, 상기 멀티미디어 템플레이트 파일 시스템의 템플레이트 파일은

각 미디어 데이터에 대한 웹서버 내의 저장위치, 매체타입, 매체이름, 표현시작시간, 표현끝시간에 대한 정보로 이루어지고,

템플레이트 파일은 클라이언트 자바 애플릿이 처음 초기화 할 때, 동기화 엔진이 참조하여 기본 동기화 전략을 수립하는 데에 사용되며,

웹서버에 의해 비디오 서버 서브릿이 구동될 때 상기 비디오 템플레이트 파일을 메모리에 로드시키는 단계를 더 포함하는 것을 특징으로 하는 자바를 이용한 동적 멀티미디어 웹 카타로그 방법.

#### 청구항 12

제10항에 있어서, 상기 비디오 데이터 스트리밍 단계는

사용자로부터 클라이언트 단말기에서 비디오 서브릿에 대한 세션 접속을 실행하여 사용자가 원하는 비디오 타이틀과 이에 관한 비디오 데이터에 대한 파라미터를 포함하는 명령어를 비디오 서버 엔진으로 전달하는 단계와,

입력된 명령어로부터 비디오 타이틀명을 파싱한 후, 파싱된 타이틀에 대한 메모리의 해당 템플레이트 객체를 선택하고 입력된 파라미터로부터 비디오 데이터를 읽어 들일 재생위치, 재생속도, 프레임율(FPS)의 변경이 있는 지를 순차적으로 판단하는 단계와,

모든 입력 파라미터에 대한 파싱 절차를 완료한후 실제 현재의 오프셋에서부터 현재의 속도로 비디오 스트림 파일을 읽어 들여 비디오 데이터가 읽히는 데로 현재의 FPS 만큼 스트림을 패킷화하여, 클라이언트 단말기로 패킷을 전송하는 단계로 구성되는 것을 특징으로 하는 자바를 이용한 동적 멀티미디어 웹 카타로그 방법.

#### 청구항 13

제10항에 있어서, 상기 카타로그 미디어 데이터 패킷 수신단계는

상기 미디어 수신 엔진 객체가 생성된 경우 현재의 네트워크 대역폭을 계산하기 위해 미리 설정한 시간만큼의 패킷을 사전에 미리 서버 시스템에 요구하여 서버 시스템으로부터 설정된 시간만큼의 패킷을 전송 받은 후 초당 평균 수신시간을 계산하는 단계와,

그후 평균 수신 시간값에 기초하여 현재의 네트워크 상태에서 최적의 FPS를 계산하는 단계와,

일단 현재 상태에서 최적의 FPS가 계산이 되면, 실제 서버 시스템으로부터 데이터가 전송될 때까지 비동기적인 리스(listen)상태에 들어가서 서버 시스템으로부터 데이터 패킷이 도착하는 지를 판단하는 단계와,

판단결과 데이터 패킷이 도달하면, 1초당의 데이터 패킷을 수신하고 다음으로 전송 받은 패킷의 헤더 부분에서 미디어 타입을 파싱하여 해당 미디어 큐에 데이터 패킷을 삽입하는 단계로 구성되는 것을 특징으로 하는 자바를 이용한 동적 멀티미디어 웹 카타로깅 방법.

#### 청구항 14

제13항에 있어서, 연속적으로 수신되는 패킷들을 미디어 큐에 삽입할 때 수신도중에 갑자기 대역폭상황이 나빠져 마지막으로 전송 받은 패킷의 논리적 끝 시간이 현재 재생중인 논리적 시간(재생위치)을 초과하는 지를 판단하는 단계와,

판단결과 마지막 패킷의 논리적 끝 시간이 현재 재생중인 논리적 시간(재생위치)을 초과하는 경우 동기화 엔진의 재생속도를 0으로 세팅하고 소정 시간동안 블록상태로 설정하는 단계와,

블록설정상태에서 초당 평균 수신값을 기준으로 최적의 버퍼링 시간을 예측하여 버퍼링 시간만큼 데이터 패킷을 수신한 후 미디어 큐에 삽입하는 단계와,

버퍼링 작업이 완료되면 다시 동기화 재생속도를 1로 세팅하여 멈췄던 지점으로부터 재생을 다시 시작하고 초당 평균 수신시간을 설정된 매 시간마다 재 계산하는 단계를 더 포함하는 것을 특징으로 하는 자바를 이용한 동적 멀티미디어 웹 카타로깅 방법.

#### 청구항 15

제10항에 있어서, 상기 미디어 데이터들에 대한 동기화 엔진의 동기화 과정에서 비디오의 각 데이터 프레임은 FPS를 기준으로 자신의 논리적 재생시간이 결정되고, 오디오는 자신의 초당 샘플링율에 따라 각 데이터 프레임의 논리적 재생시간이 결정되며, 이미지, 텍스트, HTML을 포함하는 불연속 미디어는 해당 템플릿 파일의 시작시간과, 끝시간 및 재생속도 정보를 이용하여 조합되는 미디어들의 논리적 재생시간을 결정하여 동기화가 이루어지는 것을 특징으로 하는 자바를 이용한 동적 멀티미디어 웹 카타로깅 방법.

#### 청구항 16

제10항에 있어서, 상기 미디어 데이터는 동기화 엔진의 가상 VCR 기능에 의해 재생 제어되는 것을 특징으로 하는 자바를 이용한 동적 멀티미디어 웹 카타로깅 방법.

#### 청구항 17

제16항에 있어서, 상기 동기화 엔진은 사용자로부터의 입력사항에 따른 재생속도, 재생위치 및 초당 프레임율(FPS)을 서버 시스템으로 전송한 후 각 미디어 플레이어는 현재의 동기화 상황에 대한 동기화 변수 및 플래그를 참조할 수 있도록 자신의 객체 레퍼런스, 재생속도 및 논리적 시간을 각 미디어 플레이어에게 공개하는 단계를 더 포함하는 것을 특징으로 하는 자바를 이용한 동적 멀티미디어 웹 카타로깅 방법.

#### 청구항 18

제10항에 있어서, 상기 선택된 미디어에 따라 생성되는 미디어 플레이어 객체는 자바의 멀티 스레드 기능에 의해 동시에 병렬적으로 생성되는 것을 특징으로 하는 자바를 이용한 동적 멀티미디어 웹 카타로깅 방법.

#### 청구항 19

제10항에 있어서, 사용자에 의해 상기 설정된 속도값과, 설정된 재생위치와, 설정된 초당 프레임율이 변경되는 경우 변경된 속도값, 재생위치, 초당 프레임율을 서버시스템으로 전송하는 단계를 더 포함하는 것을 특징으로 하는 자바를 이용한 동적 멀티미디어 웹 카타로깅 방법.

#### 청구항 20

제10항에 있어서, 상기 비디오 플레이어 객체의 비디오 디코딩 단계는

실제 미디어 디스플레이 엔진이 출력할 수 있는 픽셀 이미지로 변환하기 위한 비디오 디코더 객체를 스레드 객체로 생성하는 단계와,

생성된 비디오 디코더 객체에 의해 동기화 엔진의 재생 속도, 재생 위치 및 실제 비디오 프레임들 내의 논리적 시작 시간과 논리적 끝 시간을 비교하여 미디어 큐에서 현재 시각에 재생할 비디오 프레임이 있는 지를 체크하는 단계와,

상기 체크 결과 동기화 엔진의 속도가 제로이거나 또는 속도가 제로가 아니면서도 현재의 논리적 시각내에 해당하는 비디오 프레임이 없을 경우 설정된 시간만큼 자신의 스레드를 슬립하고, 만약 미디어 큐 안에 현재 논리적 시각에 해당하는 비디오 프레임이 있는 경우 그 비디오 프레임을 미디어 큐에서 가져오는 단계와,

상기 미디어 큐에서 가져온 비디오 프레임에 대하여 MPEG-1 디코딩을 수행하는 단계와,

상기 디코딩이 완료된 후 디스플레이 엔진의 비디오 디바이스를 통하여 화면에 출력시키는 단계와,

상기 출력이 현재 프레임의 재생시간 동안 화면상에 그대로 표시될 수 있도록 비디오 디코더 쓰레드를 담은 프레임 재생시간 만큼 슬립시킨 후, 출력된 비디오 프레임을 자동적으로 미디어 큐에서 제거하는 단계로 구성되는 것을 특징으로 하는 자바를 이용한 동적 멀티미디어 웹 카타로킹 방법.

#### 청구항 21

제10항에 있어서, 상기 오디오 플레이어 객체의 오디오 디코딩 단계는

실제 미디어 디스플레이 엔진이 출력할 수 있는 오디오 데이터로 변환하기 위한 오디오 디코더 객체를 쓰레드 객체로 생성하는 단계와,

생성된 오디오 디코더 객체에 의해 동기화 엔진의 재생 속도, 재생 위치 및 실제 오디오 프레임들 내의 논리적 시작 시간과 논리적 끝 시간을 비교하여 미디어 큐에서 현재 시각에 재생할 오디오 프레임이 있는 지를 체크하는 단계와,

상기 체크 결과 동기화 엔진의 속도가 제로이거나 또는 속도가 제로가 아니면서도 현재의 논리적 시각내에 해당하는 오디오 프레임이 없을 경우 설정된 시간만큼 자신의 쓰레드를 슬립하고, 만약 미디어 큐 안에 현재 논리적 시각에 해당하는 오디오 프레임이 있는 경우 그 오디오 프레임을 미디어 큐에서 가져오는 단계와,

상기 미디어 큐에서 가져온 오디오 프레임에 대하여 GSM 디코딩을 수행하는 단계와,

상기 디코딩이 완료된 후 디스플레이 엔진의 오디오 디바이스를 통하여 스피커로 출력시키는 단계와,

상기 출력이 현재 프레임의 재생시간 동안 스피커에서 그대로 재생될 수 있도록 오디오 디코더 쓰레드를 담은 프레임 재생시간 만큼 슬립시킨 후, 출력된 오디오 프레임을 자동적으로 미디어 큐에서 제거하는 단계로 구성되는 것을 특징으로 하는 자바를 이용한 동적 멀티미디어 웹 카타로킹 방법.

#### 청구항 22

제10항에 있어서, 상기 이미지 플레이어 객체의 이미지 디코딩 단계는

실제 미디어 디스플레이 엔진이 출력할 수 있는 픽셀 이미지로 변환하기 위한 이미지 디코더 객체와, 2D 이미지와 3D 이미지 모델링을 미디어 적용 타입에 따라 선별적으로 취급할 수 있는 서브 디코딩 컴포넌트 객체들을 함께 쓰레드 객체로 생성하는 단계와,

생성된 이미지 디코더 객체에 의해 동기화 엔진의 재생 속도, 재생 위치 및 실제 이미지 프레임들 내의 논리적 시작 시간과 논리적 끝 시간을 비교하여 미디어 큐에서 현재 시각에 재생할 이미지 프레임이 있는 지를 체크하는 단계와,

상기 체크 결과 동기화 엔진의 속도가 제로이거나 또는 속도가 제로가 아니면서도 현재의 논리적 시각내에 해당하는 이미지 프레임이 없을 경우 설정된 시간만큼 자신의 쓰레드를 슬립하고, 만약 미디어 큐 안에 현재 논리적 시각에 해당하는 이미지 프레임이 있는 경우 그 이미지 프레임을 미디어 큐에서 가져오는 단계와,

상기 미디어 큐에서 가져온 이미지 프레임에 포함된 이미지 템플레이트 파일의 이미지를 가져올 미디어 파일 디렉토리, 이미지 타입, 논리적 시작시간 및 논리적 끝시간 정보를 바탕으로 실제의 이미지 데이터를 서버 시스템의 웹서버로부터 실시간으로 읽어와 논리적 재생시간을 결합한 완전한 이미지 프레임을 완성하는 단계와,

상기 완성된 미디어 프레임이 20인지 30인지를 판단하여 20인 경우 미디어 프레임 조각들에 대하여 20 이미지 디코딩을 수행하고, 30인 경우 3D 이미지 모델링을 수행하는 단계와,

상기 디코딩이 완료된 후 디스플레이 엔진의 비디오 디바이스를 통하여 화면에 출력시키는 단계와,

상기 출력이 현재 프레임의 재생시간 동안 화면상에 그대로 표시될 수 있도록 이미지 디코더 쓰레드를 담은 프레임 재생시간 만큼 슬립시킨 후, 출력된 이미지 프레임을 자동적으로 미디어 큐에서 제거하는 단계로 구성되는 것을 특징으로 하는 자바를 이용한 동적 멀티미디어 웹 카타로킹 방법.

#### 청구항 23

제22항에 있어서, 상기 3D 이미지 모델링은 서로 다른 각도에서 촬영된 18장 이상의 연속적인 2D 이미지들을 순차적으로 받아들이면서 3차원적으로 모델링해주는 모션 모듈과, 한장의 2D 이미지를 가지고 3차원적인 360도 패닝 효과를 구현함과 동시에 이미지 내의 특정 공간을 클릭했을 때 다른 미디어 플레이어와 연동시켜주는 기능을 포함하는 파노라마 모듈 중 어느 하나로 이루어지는 것을 특징으로 하는 자바를 이용한 동적 멀티미디어 웹 카타로킹 방법.

#### 청구항 24

제10항에 있어서, 상기 웹 카타로킹을 동적으로 수행할 때 각각의 미디어는 독자적 재생, 하나 이상의 동일 및/또는 서로 상이한 미디어 조합 중 어느 하나의 형태로 미디어가 재생되며,

상기 재생되는 미디어 조합은 비디오와 텍스트 자막으로 이루어진 제1미디어 조합, 비디오와 오디오로 이루어진 제2미디어 조합, 비디오, 오디오 및 텍스트 자막으로 이루어진 제3미디어 조합, 비디오, 오디오, 텍스트 자막 및 스틸 이미지로 이루어진 제4미디어 조합, 비디오, 오디오, 제1언어 텍스트 자막 및 제2언어 텍스트 자막으로 이루어진 제5미디어 조합, 비디오, 오디오, 제1언어 텍스트 자막, 제2언어 텍스트 자막 및 제3언어 텍스트 자막으로 이루어진 제6미디어 조합, 비디오, 오디오 및 HTML 문서로 이루어진 제7미디어 조합, 비디오, 오디오, 텍스트 자막 및 HTML 문서로 이루어진 제8미디어 조합, 비디오, 오디오, 텍스트 자막, HTML 문서 및 스틸 이미지로 이루어진 제9미디어 조합, 스틸 이미지 및 텍스트 자막으로 이루어져서 슬라이드 쇼를 구현하는 제10미디어 조합, 스틸 이미지, 텍스트 자막 및 각 이미지 당



오디오로 이루어져서 오디오 슬라이드 쇼를 구현하는 제11미디어 조합, 애니메이션 이미지에 의해 롤링 배너광고를 구현하는 제12미디어 조합, 애니메이션 이미지 및 각 이미지 당 오디오로 이루어져서 사운드 롤링 배너 광고를 구현하는 제13미디어 조합, 3D 파노라마 이미지의 제14미디어 조합, 다른 각도에서 찍은 2D 이미지의 연속에 의해 3D 사진 모델링 효과를 구현하는 제15미디어 조합, 애니메이션 이미지, 텍스트 자막 및 각 이미지 당 오디오로 이루어져서 사운드/문자 롤링 배너 광고를 구현하는 제16미디어 조합 중 어느 하나의 조합인 것을 특징으로 하는 자바를 이용한 동적 멀티미디어 웹 카타로킹 방법.

#### 청구항 25

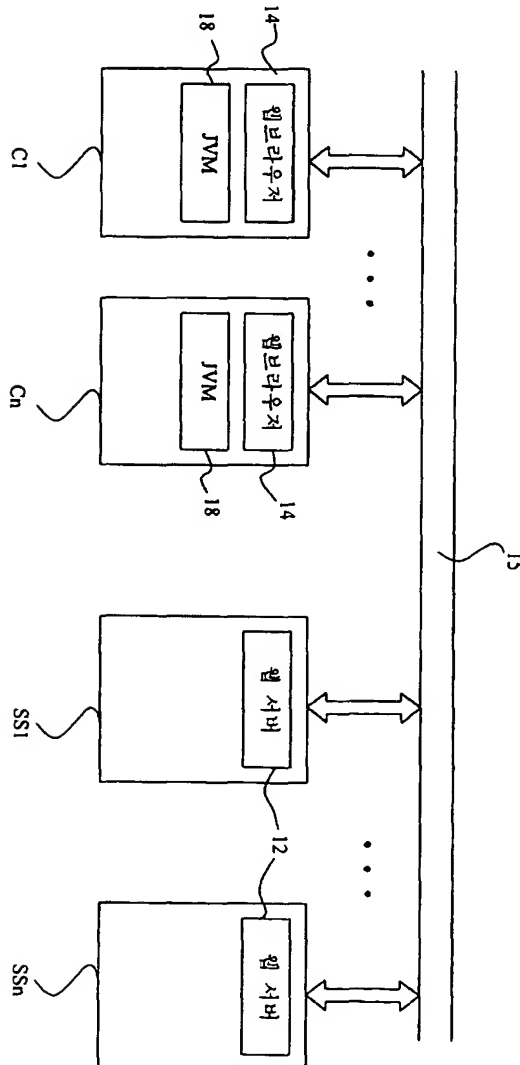
제24항에 있어서, 상기 비디오는 MPEG-1 데이터이고, 오디오는 실시간 GSM 데이터인 것을 특징으로 하는 자바를 이용한 동적 멀티미디어 웹 카타로킹 방법.

#### 청구항 26

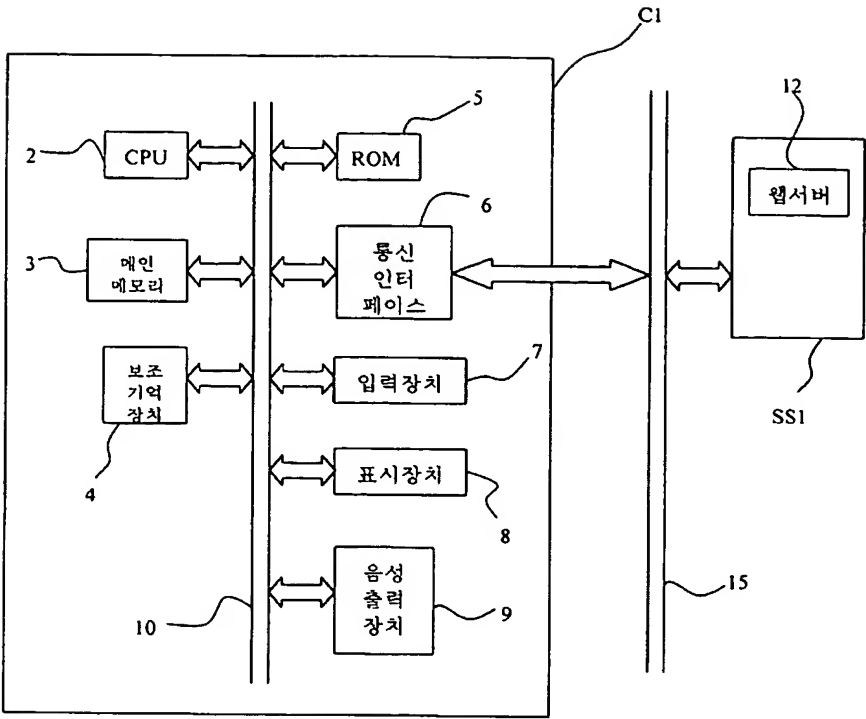
제10항에 있어서, 상기 클라이언트 단말기는 순수 자바 기반으로 설계된 자바 버추얼 머신과 웹 브라우저를 탑재한 PC(개인용 컴퓨터), 웨스테이션, 노트북 PC 및 팜탑 PC와, 퍼스널 자바를 포팅한 일반 휴대용 통신단말기와, 휴대 정보 단말기(PDA)와, 셋톱 박스와, 디지털 텔레비전과, 웹폰과, 자바 오퍼레이팅 시스템(JavaOs)을 탑재한 자바 칩 중 어느 하나인 것을 특징으로 하는 자바를 이용한 동적 멀티미디어 웹 카타로킹 방법.

#### 도면

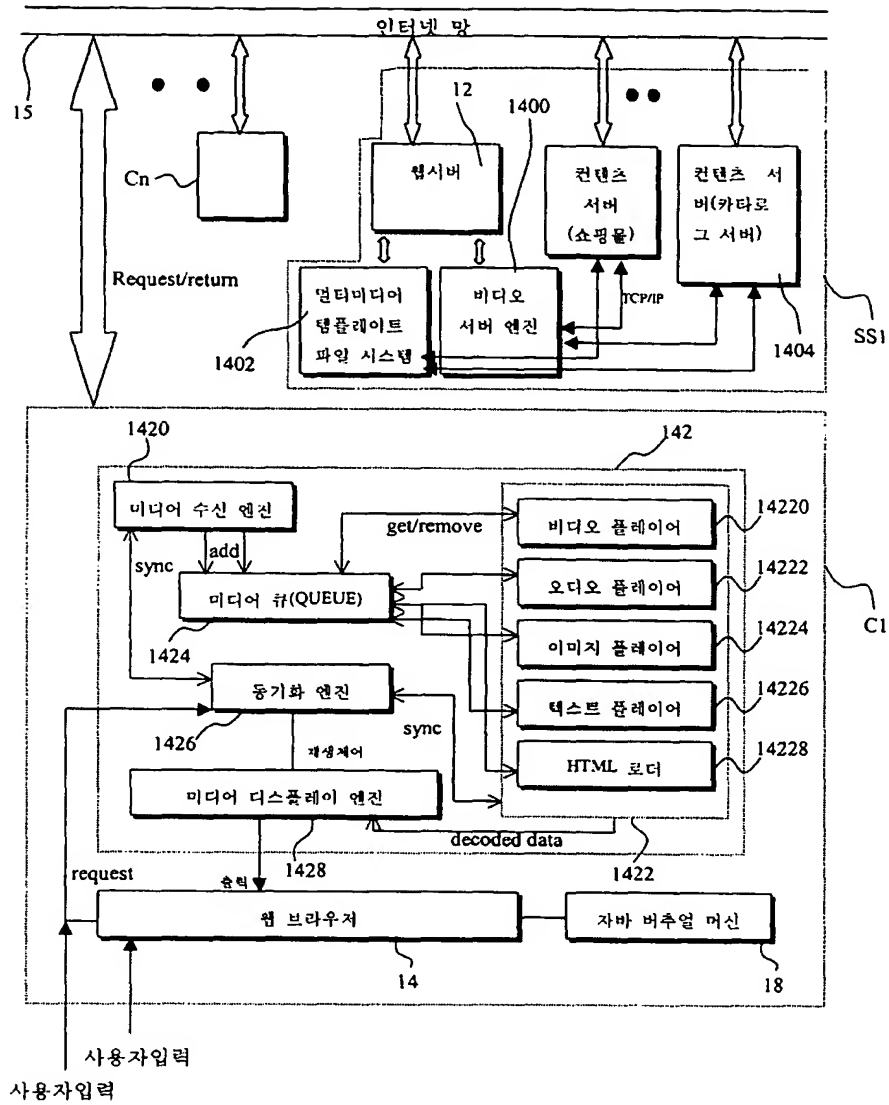
도면1



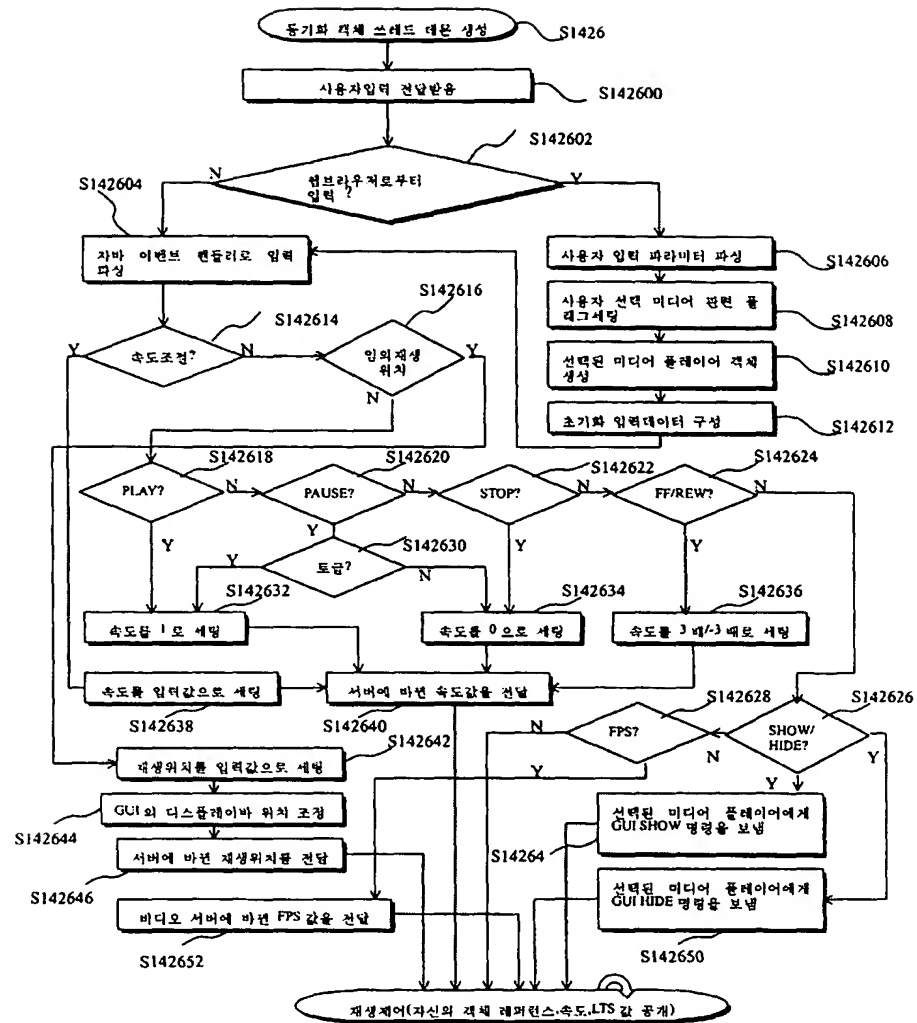
도면2



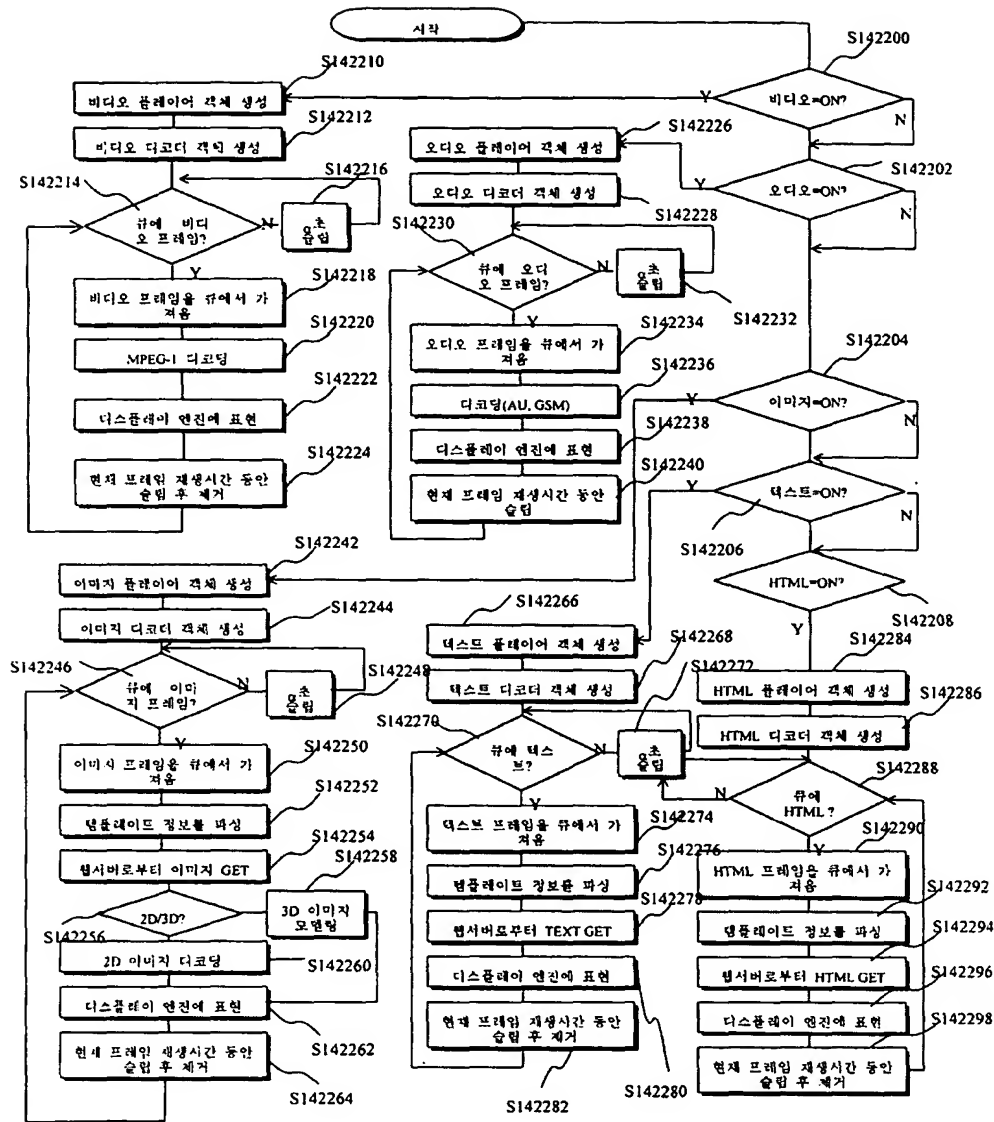
도면3



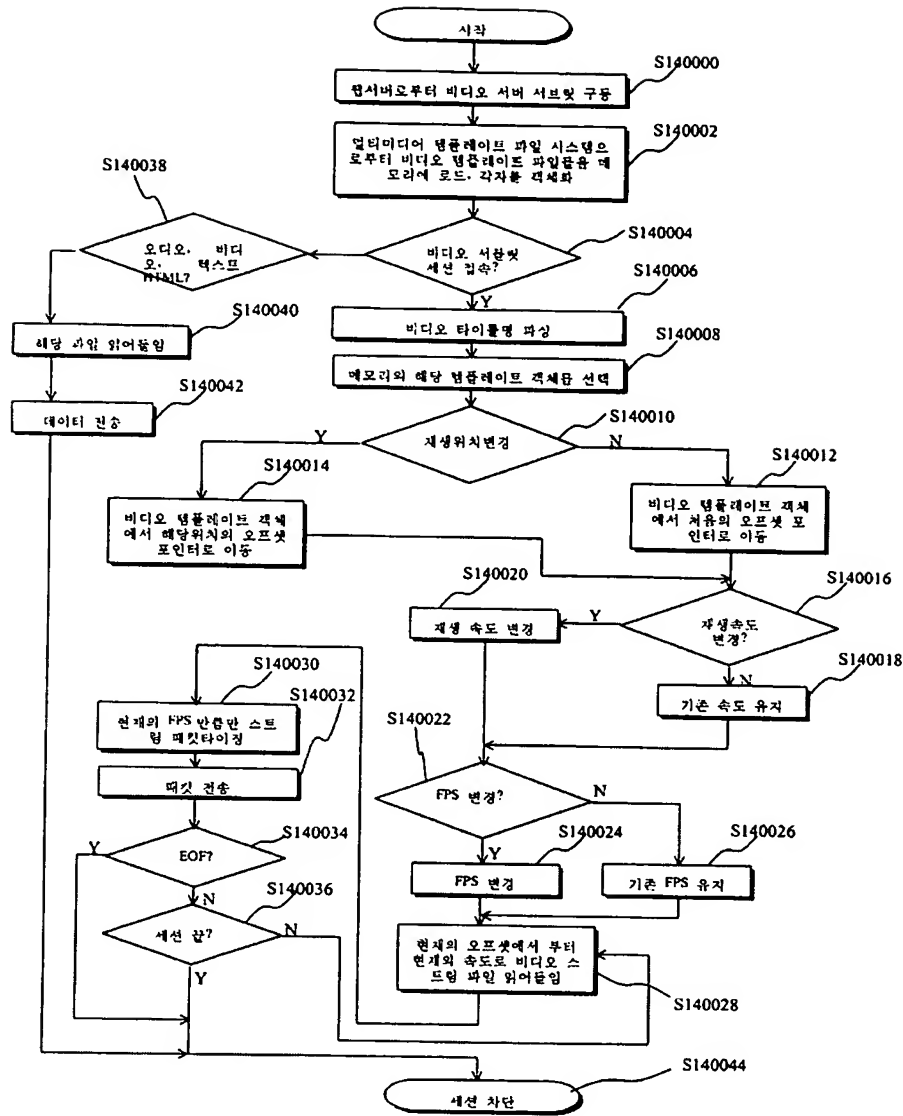
도면4



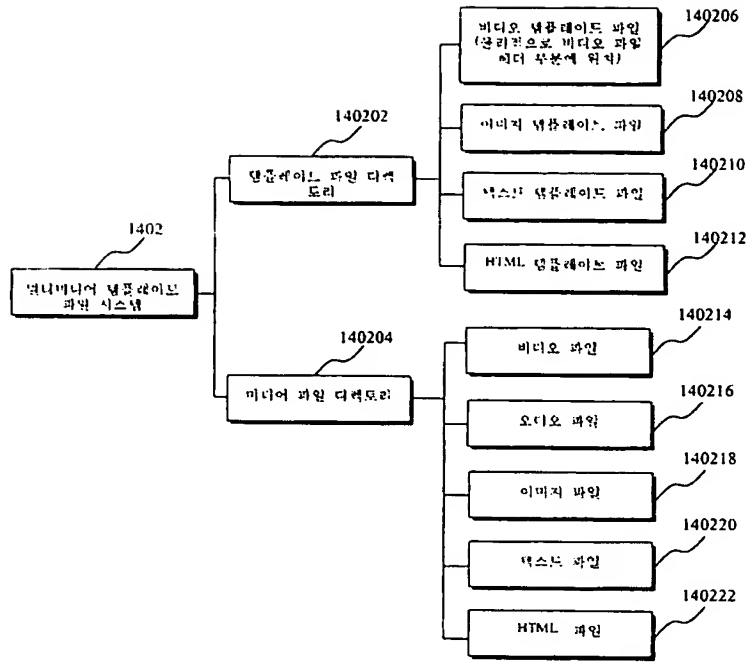
도면5



도면6

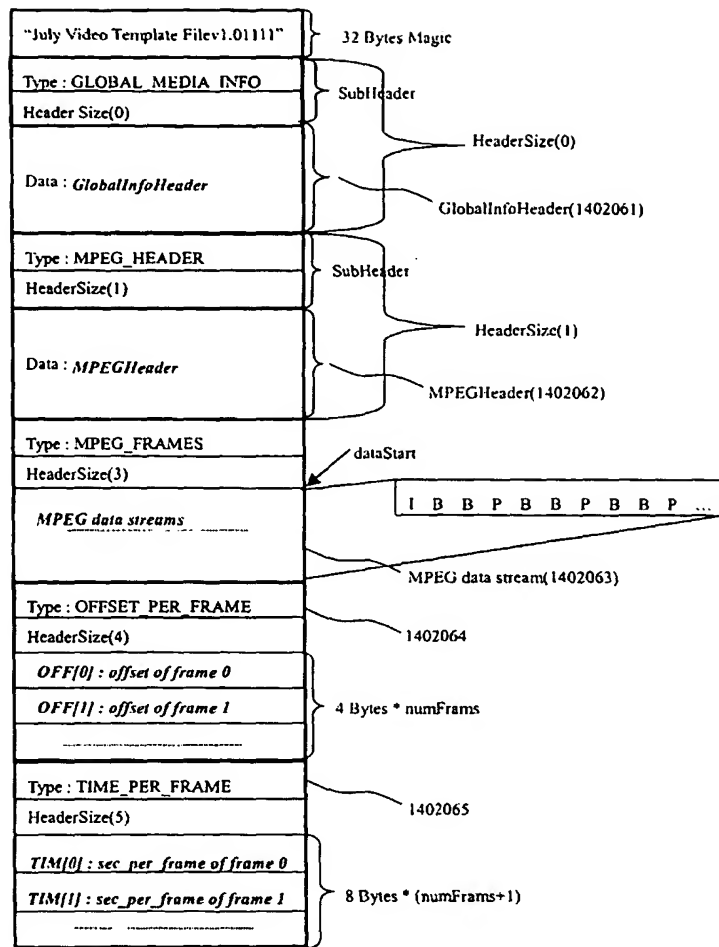


도면7

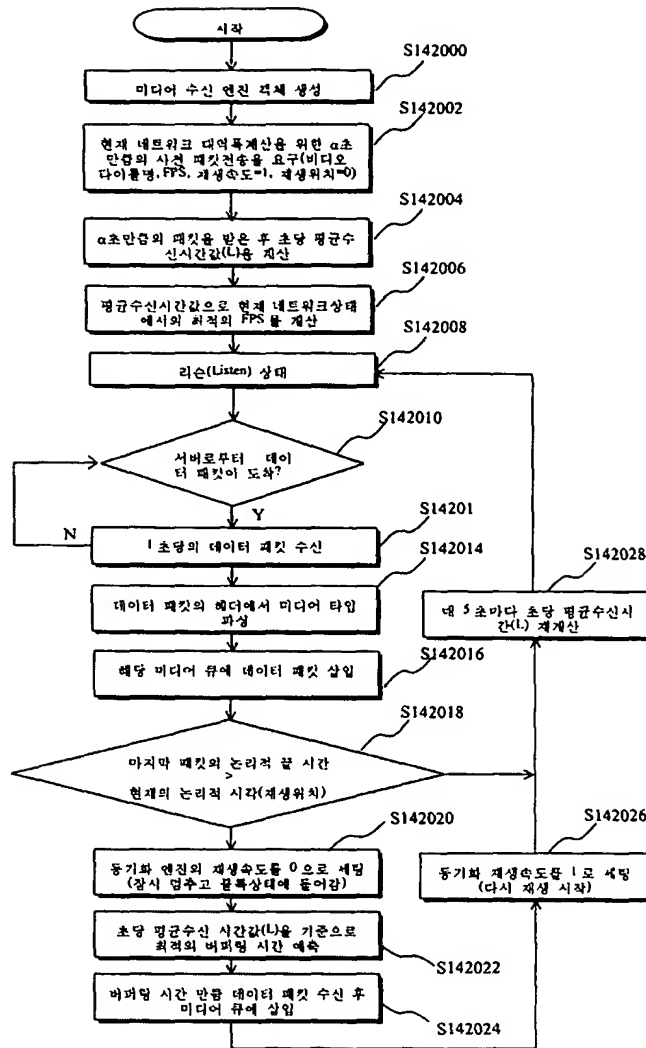




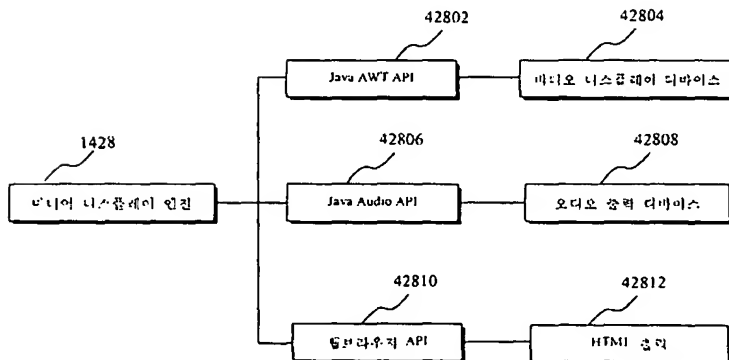
도면8



도면9



도면10



도면 11

